

MAGAZINE
MSX
AÑO I
Núm. 12
Abril 1986
300 Ptas.
Test:
Toshiba
HX-20
Aplicación:
sistemas de
ecuaciones II
Código
máquina
Libros,
Programas,
Software...
SVI-328 PRECURSOR DEL ESTANDAR

MSX

DYNADATA

NO LO DUDE



PRECIO INSUPERABLE

CONJUNTO:
ORDENADOR
MONITOR Y
CASSETTE
DYNADATA MSX

62.500

TARJETA
PROGRAMA.
EL NUEVO FORMATO DEL FUTURO.

MONITOR 12"
FOSFORO
VERDE

ADAPTADOR TARJETA

DATA CASSETTE

NUEVO

DYNADATA MSX
CON TECLADO EN
ESPAÑOL*

* Letras y signos iguales al teclado del PC de IBM.

Ordenador DYNADATA MSX con teclado en español	**46.900**
DYNADATA MSX con cassette y monitor de color	**95.500**
DYNADATA MSX y unidad de diskette de 5 ¼" de 360 Kbytes con monitor de fósforo verde	**108.900**
con monitor de color	**141.000**
TARJETA/PROGRAMA del juego LE MANS con adaptador, el cual sirve para cualquier tarjeta que usted adquiera. Precio especial con la compra del DYNADATA MSX	**4.900**

Con el DYNADATA MSX usted podrá:

- Divertirse con la amplia gama de juegos MSX.
- Aprender Informática y Basic con el curso autodidáctico y audiovisual.
- Llevar gestiones administrativas con los programas de proceso de textos, base de datos, contabilidad, stock, recibos, etc.
- Ayudar a sus hijos en sus estudios de 3.º a 8.º de EGB con los programas de Matemáticas, Lenguaje y Ciencias Naturales.
- Aprender idiomas tan necesarios como el inglés.
- Programar con los lenguajes: LOGO, PASCAL, FORTRAN y COBOL.

CURSO DE INFORMATICA Y BASIC

- Autodidáctico
- Audiovisual
- 12 cassettes
- 24 lecciones
- Evaluaciones periódicas
- Diploma Fin de Curso



DYNADATA MSX Y CURSO **69.300**

Especificaciones DYNADATA MSX: Procesador Z80A, 64 Kbyte RAM, 16 Kbyte VRAM, 32 Kbyte ROM, 24 líneas x 40 columnas, 256 x 192 pixels, 16 colores, MSX-BASIC, MSX-DOS.

DYNADATA

Y:

	PIONEER
SONY	YAMAHA
PHILIPS	MITSUBISHI
CANON	GOLDSTAR
SANYO	SAMSUNG
JVC	HITACHI
TOSHIBA	MATSUSHITA
SPECTRAVIDEO	CASIO

Se han decidido por MSX. Esto le permite compartir los programas y periféricos con todas estas reconocidas marcas.

Por todo, NO LO DUDE. Decídase por



Solicite información: Sor Angela de la Cruz, 24 - 28020 Madrid. Tels. (91) 279 21 85 - 279 28 01 - 270 01 93. Telex 44619 DYNA
Delegación Barcelona: Aribau, 61, entlo - 08011 Barcelona. Tels. (93) 254 73 04 - 254 73 03



EDITORIAL

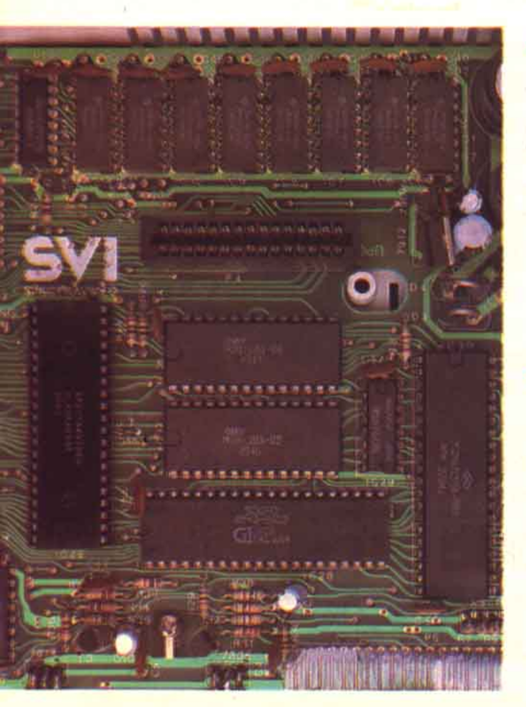


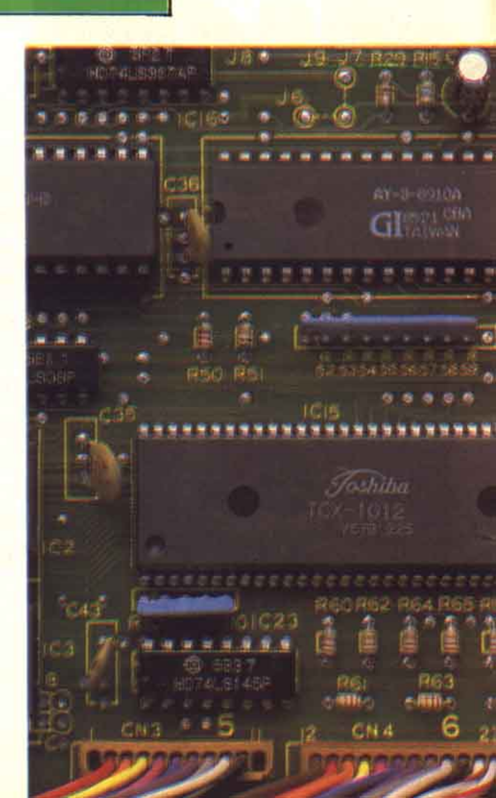
Hace un año que **MSX Magazine** salió a la calle con la ilusión y las ganas de cubrir un lugar en el mundo editorial, donde faltaba una publicación que se dedicara al estándar MSX. Pues bien, aquí estamos una vez más, con un tema de portada sugerido por la gran cantidad de cartas recibidas en esta redacción; el SV-328. Los asiduos a la publicación, se preguntarán por qué un ordenador que no es MSX aparece como el tema principal de este mes. Está claro que durante el desarrollo de cualquier máquina o actividad, el hombre tiende a olvidarse repentinemente de las raíces de que partió ese desarrollo, y nosotros no hemos sido menos. El SV-328, fue el precursor del MSX, del cual se fueron tomando las ideas básicas (teclado, port de cartuchos, etc.) y se complementó con los interfaces adecuados para convertirlo en el SV-728, ordenador conocido por nuestros lectores.

Por este motivo en este número, con el que cumplimos un año, rendimos un pequeño homenaje a dicho ordenador, a los usuarios que lo poseen y aún lo disfrutan. Por si esto fuera poco, tenemos que destacar y resaltar que POR FIN ya están aquí. La 2.ª generación ha llegado de la mano de Philips, que ha presentado a mediados del mes pasado su VG-8235, un aparato que rompe el hielo creado por la misteriosa generación y de la que sólo hemos podido ofrecer literatura y nada de hechos. Pues bien, los tan deseados 128k de MSX ya están aquí para competir con el apretado mercado de ordenadores de esta talla. Recordemos que actualmente existen varios modelos de las casas más renombradas con ordenadores de 128k y que la competencia será dura, aunque las características de estos nuevos aparatos dista mucho de las que ofrecen sus competidores. Esperemos poder ofrecerles la primera crítica de estos nuevos ordenadores dentro de poco tiempo.


SVI
605 EXPANDER
SVI 328
POWER ON

SUMARIO


SVI


AY-3-8910A
Toshiba
TCX-1012







NOTICIAS



Ha sido constituida la Sociedad SVI España

Por fin los sufridos usuarios de Spectravídeo contarán con una empresa dedicada a ellos. Ya no tendrán que clamar en el desierto (sinónimo de Indescomp), ni tendrán que recorrerse todas las tiendas de su localidad para que, por lo menos, le hagan caso a uno. A eso se ha llegado después de que Stephen Chu, Director de Spectravídeo International LTD. y D. Carlos García-Soto, Director Gerente de SVI España S.A., firmasen el acuerdo por el que esta empresa comercializará en exclusiva los productos Spectravídeo en España y Portugal.

Este paso, viene a ser el último eslabón de la cadena de compañías que, en los distintos países europeos, integran el grupo SVI Europe.

El gran atractivo de esta empresa es que, no sólo se van a dedicar a la comercialización de los ordenadores domésticos de esta misma marca en los citados países, sino que también existe la posibilidad de comercializar juguetes y periféricos.

El acuerdo recoje la próxima fabricación de algunos de los productos mencionados anteriormente en nuestro país, con la posibilidad de poder suministrarlcs a otros países.

La nueva empresa tiene planes muy ambiciosos para el mercado español, que se materializarán, a corto plazo, debido al empuje renovado y directo, ejercido sobre la gama de productos SVI.

Para obtener más información, dirigirse a la sede de SVI España, situada en la Avda. de la Constitución, 260, en Torrejón de Ardoz, Madrid. Tel.: 675 75 99 y Télex 48099 Proto E.

El final feliz de esta historia la encontraremos en la explosiva bajada de precios que estos productos han tenido, ofreciéndose, el Spectravídeo X'-press a 84.400 ptas. (cuando antes se comercializaba a 99.900 ptas.), el Spectravídeo 728 a 39.900 ptas., por citar dos ejemplos.

Nueva publicación para los amantes de la música

La utilización de la tecnología electrónica en las artes del registro de sonidò y la composición mucical no es nada nuevo. Muchos de los conceptos teóricos y prácticos en que se fundamentan los actuales medios de creación y manipulación del sonido se remontan al principio mismo del siglo.

No obstante, lo que sus inicios fue una tecnología notablemente com-

pleja, a la que sólo tenían acceso los investigadores implicados y un reducido sector de artistas, se ha convertido hoy, cruzada la frontera de los 80, en una innumerable lista de marcas, técnicas, sistemas y modelos que ofrecen toda una gama de sofisticados instrumentos y equipos musicales, con unas posibilidades y características impensables, hace tan sólo cinco años.

Tenemos la suerte de vivir en una época trascendente para la música y las artes audiovisuales, época en la que afloran las más fecundas y revolucionarias ideas en cuanto a medios e instrumentos.

MUSICA Y TECNOLOGIA nace con estos telones de fondo: con un sector artístico cuyas principales estrellas se sirven habitualmente del medio electrónico para sus creaciones, con un mercado en el que es posible adquirir instrumentos por precios, que oscilan desde las cinco mil pesetas hasta varios millones, con unas técnicas de manipulación sonora que hacen posible orquestar cualquier ruido imaginable.

Ya nadie se conforma con el simple ejercicio de la conjetura en torno a las maravillas de la técnica. MUSICA Y TECNOLOGIA te ofrece la posibilidad de conocerlas, tocarlas, disfrutarlas y, sobre todo, entenderlas.

Primer ordenador de la 2.ª Generación

La tan sonada 2.ª generación ha hecho su entrada de la mano de Philips, cuyo modelo VG-8235, se ha presentado hace poco, con las destacadas posibilidades que en su momento comentamos (ver MSX Magazine de marzo). Con la introducción de estos ordenadores, Philips ha dado un paso importante en el desarrollo y ampliación lógica de la gama global del sistema MSX.

Por otro lado, esta empresa ha sido elegida como uno de los tres fabricantes de sistemas informáticos que serán recomendados por la FERE (Federación Española de Religiosos de Enseñanza) para su proyecto de informatización de las aulas de estos colegios. El denominado plan «ALFA-FERE».

Para ello se han organizado en diez puntos distintos de la geografía española ciclos de conferencias y aulas de demostración en las que se espera que participen las distintas instituciones que están afiliadas a esta Federación.



Redada en el Rastro

Durante la mañana del domingo 2 de marzo, el grupo 7.° de la Brigada Regional de la Policía Judicial, con apoyo del grupo 8.° y de la Comisaría de Arganzuela, han precedido a la retirada de cintas con programas para microordenador que se vendíian de forma ilegal en el conocido Rastro madrileño.

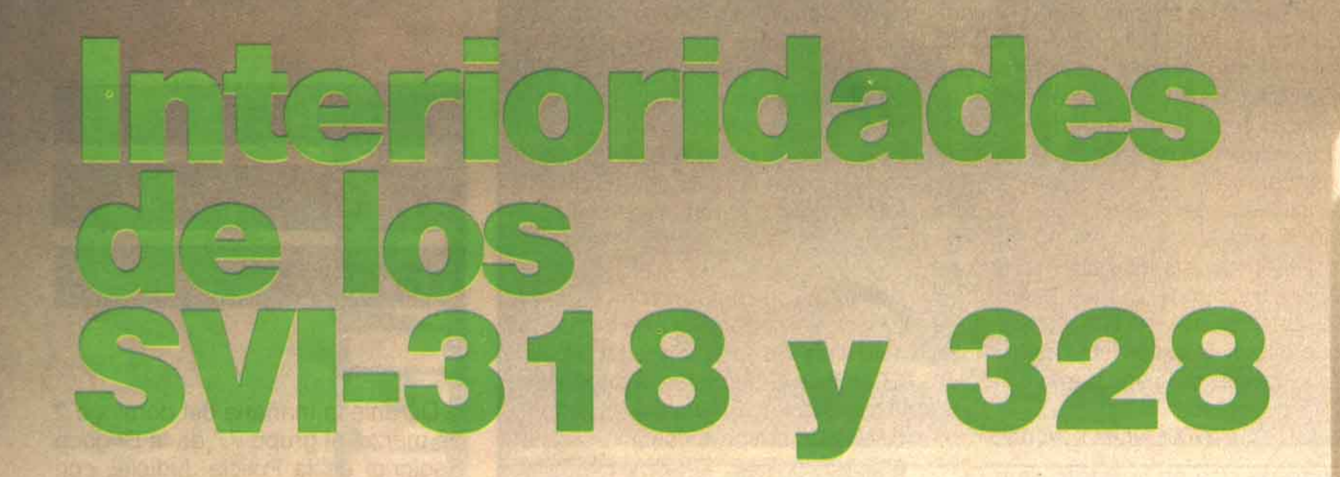
La denuncia había sido presentada días antes por ANEXO, Asociación Española de empresas fabricantes de software para ordenadores profesionales y domésticos, ante la creciente actividad de los piratas de programas.

El número de programas retirados por la Policía en la mañana del domingo supera las 11.000 unidades, con un valor en el mercado de más de 22 millones de pesetas.

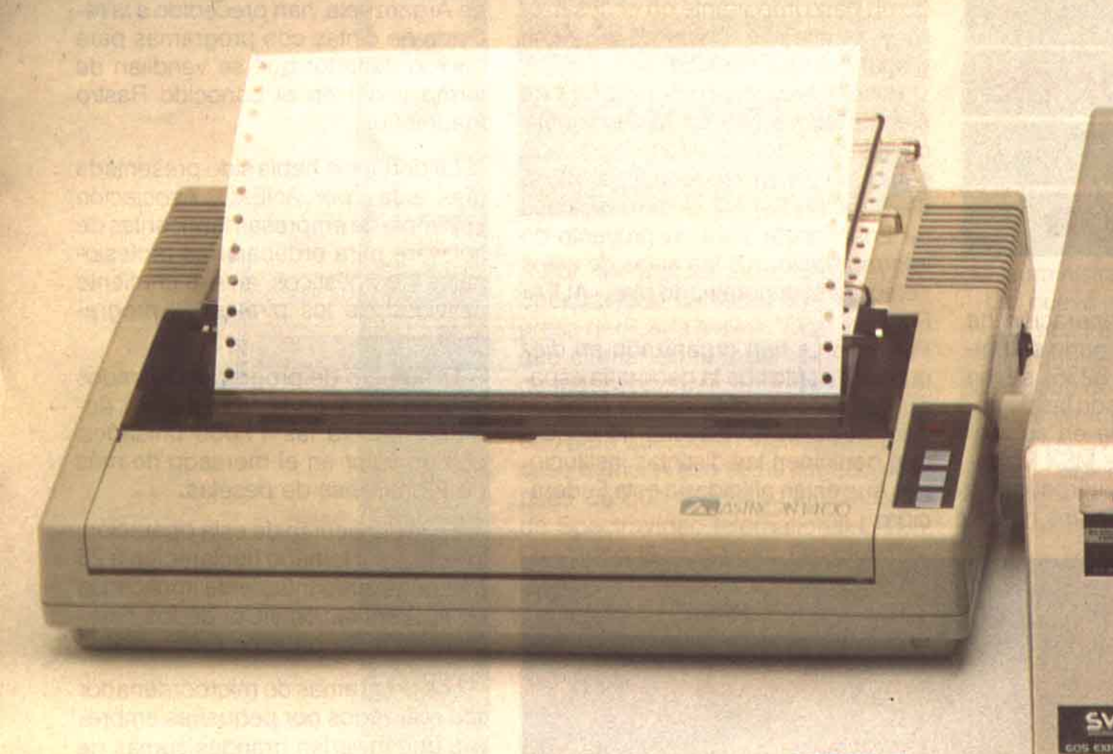
En el transcurso de esta operación, la Policía ha tomado declaración a 28 personas, presuntamente implicadas en la apropiación ilícita de los derechos de autor de esos programas.

Los programas de microordenador son realizados por pequeñas empresas que invierten grandes sumas de dinero en su creación y comercialización, y cuya viabilidad depende de conseguir vender un gran número de copias a bajo precio. La piratería, practicada a veces por grandes grupos de actividad clandestina, amenaza con extinguir en nuestro país la naciente industria del software, de enorme importancia para la futura independencia tecnológica de España.

Cabe destacar la total discrección policial, que ha evitado el mínimo desorden y las molestias para quienes visitaban esta mañana el conocido mercadillo madrileño.



Interioridades de los SVI-318 y 328



Este ordenador, precursor de la norma MSX, fue la pieza clave del desarrollo del estándar, sin embargo, la literatura sobre este ordenador ha sido más bien poca. Con este artículo, iniciamos una serie sobre este interesante equipo.

Los más introducidos en estas cosas de la informática ya saben que el *Z-80* es el chip microprocesador usado por muchos de los pequeños ordenadores actuales como cerebro electrónico.

Los ordenadores SPECTRAVIDEO, y entre ellos los *SVI-318* y *SVI-328* también están en la fami-


MicroPro
DATASTAR
WORDSTAR
REFERENCE MANUAL
CONTRAST
BRIGHT
POWER
ENTER
SHIFT

lia de los ordenadores personales cuya unidad central de proceso es el *Z-80*.

El *Z-80* es un procesador de 8 *bits*, pero puede manejar también números de 16 *bits*.

El mayor número representable con 16 *bits* es 65.536, es por esto que el número de posiciones de memoria que el *Z-80* puede direccionar en una operación de lectura o escritura en la memoria es 65.536, que equivale a un tamaño de memoria de 64 *Kbytes* ó 64 *K*.

Tanto el *SVI-318* como el *SVI-328* pueden llegar a tener 160 *Kbytes* de memoria *RAM*, más 16 *Kbytes* de *VRAM* o memoria de vídeo, más la posibilidad de manejar hasta 64 *Kbytes* de memoria *ROM* en forma de cartuchos introducidos en el «*slot*» que presenta el ordenador en su parte superior, más la *ROM* inicial.

Así pues, nos encontramos que al *Z-80*, capaz de direccionar 64 *Kbytes*, le han asignado la tarea de manejar 272 *Kbytes*, veamos cómo lo hace.

El acceso a la memoria

Empecemos por la *VRAM* o *RAM* de vídeo.

Esta parte de la memoria no se considera a efectos de la *UCP* (unidad central de proceso) como parte de la memoria principal, sino que la *UCP* la maneja como si de otro periférico se tratara, enviando y recibiendo datos y direcciones a través de los *PORTS* que la comunican con la mencionada *VRAM* y con el procesador de vídeo. La interpretación que el procesador de vídeo hace de estos datos, la veremos en el capítulo del próximo número.

El resto de la memoria se va alojando, a medida que se va instalando, en 8 bancos de memoria cuya representación puede verse en el apéndice E del manual de usuario proporcionado con el ordenador, o en la figura 1 de este texto.

DIRECCION				
0000 hex 7FFF hex	banco 01 ROM incluida	banco 11 ROM cartucho	banco 21 RAM SVI-328	banco 31 RAM ampliada
8000 hex BFFF hex COOO hex FFFF hex	banco 02 RAM SVI-328 RAM incluida	banco 12 ROM cartucho	banco 22 RAM ampliada	banco 32 RAM ampliada

Figura 1. Bancos de memoria en los SVI-318 y 328.

El banco 01 contiene los 32 *Kbytes* de*ROM* inicial que alberga al sistema operativo, el traductor de *BASIC* y la relación de códigos de carácteres y comandos, tanto en el *SVI-318* como en el *SVI-328*.

La memoria se puede dividir en cuatro participaciones o bancos, que son el 01, 11, 21 y 31.

El banco 02 contiene 32 *Kbytes* de memoria *RAM* en el caso de los *SVI-328* y 16 *Kbytes* en los *SVI-318* sin ampliar, ya que los otros 16 *Kbytes* que completan la memoria *RAM* del *SVI-318* forman la *RAM* de vídeo.

Los bancos 11 y 12 se refieren a los cartuchos *ROM*.

El banco 21 aloja 32 *Kbytes* de *RAM* incorporada en el caso de los *SVI-328* ó en el *SVI-318* expandido.

Los restantes bancos: 22, 31 y 32, son los lque recibirán las futuras ampliaciones de memoria *RAM*.

Estos 8 bancos se clasifican en dos grupos:

— El primero comprende aquellos bancos cuyo número de denominación termina en 1, incluye los bancos 01, 11, 21 y 31.

— El segundo grupo tiene las denominaciones terminadas en 2 y comprende los bancos 02, 12, 22 y 32.

El *Z-80* tiene la facilidad de poder conmutar los bancos mediante los *PORTS 88 hex, 90 hex* y *8C hex* que le comunican con los canales A y B del generador de sonido programable además de con los registros del mismo.

En el programa 1 se comenta en detalle la forma de acceder al banco 21 en los *SVI-328*.

La conmutación de bancos permite al *Z-80* direccionar a un tiempo dos de los ocho bancos de que dispone el ordenador, con la particularidad de que uno de ellos ha de pertenecer al primer grupo, y el otro al segundo.

El banco perteneciente al primer grupo tendrá asignadas sus direcciones de memoria como



Foto 1. Red local de comunicaciones Spectravideo MSX.

comprendidas entre la dirección pandido la memoria mediante el comando *SWITCH.*

Los bancos 31 y 32 sólo son accesibles desde el código máquina, cuando tengan incorporada la correspondiente expansión de memoria.

Programa 1
Conmutación de bancos

Descripción:

Este programa conmuta virtualmente el banco 02 por el banco 21 de los *SVI-328,* para su uso desde el *BASIC.*

En realidad esto no puede suceder así, ya que, como hemos visto en el texto anterior, un sólo banco de cada grupo puede estar siendo direccionado a la vez.

Lo que se hace en realidad es intercambiar el banco 01 por el 21, de tal forma que la *UCP* está accediendo al banco 21 para las direcciones entre *0000 hex* y *7FFF hex* y al banco 02 para las direcciones *8000 hex* a *FFFF hex.*

A continuación intercambia el contenido de los bancos 02 y 21 y finalmente se retorna al estado incial con la parte *RAM* de memoria conteniendo la información del segundo banco.

La forma de acceder a la conmutación de bancos en los *SVI-318* y *328* es enviando mediate los *PORTS 88 hex* y *8C hex* un código al canal B del procesador de sonido.

El comando de conmutación de bancos es *SWITCH,* que se consigue poner en funcionamiento gracias a un «*HOOK JUMP*» o salto de enganche que existe en las variables del sistema y que es llamado desde la rutina *ROM* que interpreta los comandos. Más información sobre este y otros puntos de enganche que existen la veremos en un capítulo dedicado a las rutinas de la memoria *ROM.*

El programa en *BASIC* es un cargador que introduce un programa que consta de dos partes:

— La primera es una rutina que prepara al segundo banco para su *0000 hex* y la dirección *7FFF hex,* y el perteneciente al segundo grupo, tendrá asignadas las direcciones *8000 hex* a *FFFF hex.*

Los bancos inicialmente conmutados al encender el ordenador son el 01 y el 02, a no ser que inicialmente el aparato con el sistema operativo *CP/M,* ya que el *CP/M* conmuta automáticamente el banco 21, en lugar del banco 01.

El banco 22 será accesible desde el *BASIC* cuando hayamos exmanejo desde el *BASIC.* Para ello introduce todos los datos contenidos en la memoria *RAM,* incluidas las propias rutinas de conmutación, en el segundo banco.

— La segunda intercambia la información de los dos bancos cada vez que es llamada mediante el comando *SWITCH.*

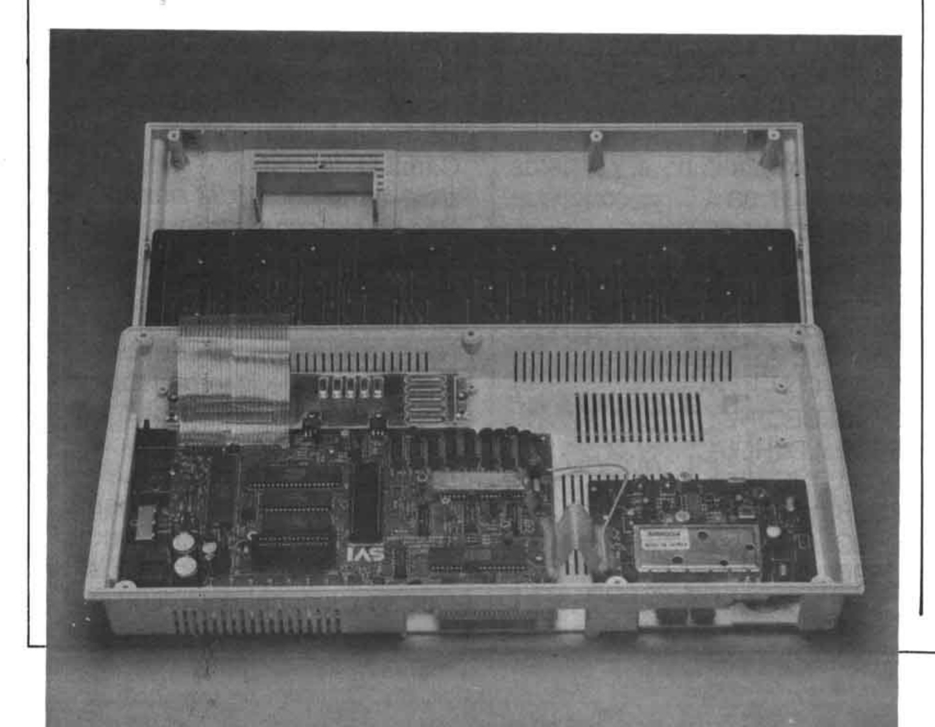


Foto 2. Aspecto interno del SVI-328.

Foto 3. Generador de Sonidos, port de expansión y microprocesador Z-80.

Programa ensamblador:

Rutina de inicialización del segundo banco

D54E...PUSH HL	Reserva retorno al *BASIC.*
D544F...LD A,C3	Coloca la dirección de
D551...LD HL,D578	comienzo de la rutina de
D554...LD (FF57),A	conmutación en el salto
D557...LD (FF58),HL	de enganche.
D55A...LD (FF5C),SP	Reserva puntero de la pila.
D55E...DI	Deshabilita interrupciones.
D55F...LD A,OF	Canal B del *PSG.*
D561...OUT 88,A	enviado a través del *PORT 88 hex.*
D563...LD A,DD	Código de conmutación
D565...OUT 8C,A	enviado a través del *PORT 8C hex.*
D567...LD HL,8000	En *HL* el comienzo del banco 02.
D56A...LD DE,0000	En *DE* el comienzo del banco 21.
D56D...PUSH HL	Dato en *HL* pasado
D56E...POP BC	al registro *BC.*
D56F...LDIR	Datos del banco 02 al banco 21.
D571...LD A,DF	Código de conmutación
D573...OUT 8C,A	a travé del *PORT 8C hex.*
D575...EI	Habilita interrupciones.
D576...POP HL	Recupera retorno al *BASIC.*
D577...RET	Retorno al *BASIC.*

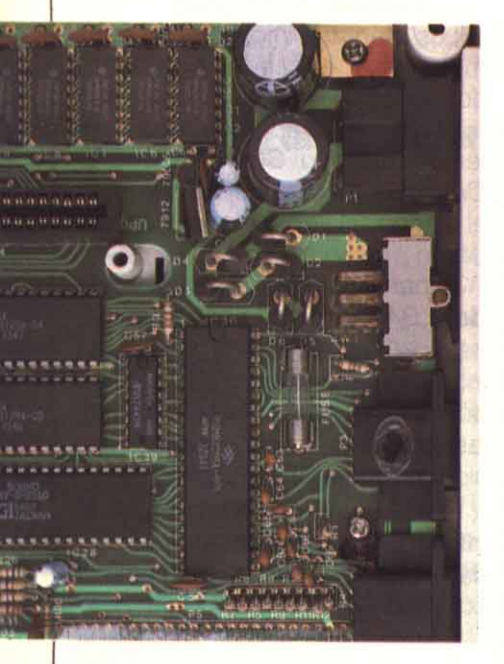


Foto 4. Fusible de protección y conectores de joysticks.

Rutina de conmutación

D578...CP C9	Prueba si comando *SWITCH.*
D57A...RET NZ	Retorna si no.
D57B...LD (FE5C),SP	Reserva puntero de la pila.
D57F...LD IX,(FA03)	Guarda la línea del cursor.
D583...LD (FE5E),HL	Reserva retorno al *BASIC.*
D586...DI	Deshabilita interrupción.
D587...LD A,OF	Canal B del *PSG*
D589...OUT 88,A	a través del *PORT 88 hex.*
D58B...LD A,DD	Codígo de conmutación
D58D...OUT 8C,A	a través del *PORT 8C hex.*
D5IF...LD HL,8000	Inicio del banco 02.
D592...LD DE,0000	Inicio del banco 21.
D595...LD A, (DE)	Reserva el contenido de *DE.*
D596...LDI	Transferencia e incremento.
D598...DEC HL	Se deposita el dato
D599...LD (HL),A	guardado en A.
D59A...INC HL	*HL* toma el siguiente valor.
D59B...LD A,H	Prueba si *HL*
D59C...OR L	es igual a 0.
D59D...JR NZ, F6	Si no, vuelta a transferir datos.
D59F...LD A,DF	Código de conmutación en A

D5A1...OUT 8C,A .... a través del *PORT 8C hex.*
D5A3...EI .... Habilita interrupciones.
D5A4...LD SP, (FE5C) .... Recupera posición de la pila.
D5A8...CALL 3750 .... Recupera color anterior.
D5AB...LD (FA03),IX .... Recupera línea del cursor.
D5AF...LD HL, (FE5E) .... Recupera retorno al *BASIC.*
D5B2...INC HL .... *HL* apunta al siguiente comando.
D5B3...LD A, (HL) .... Código del comando en A.
D5B4...POP BC .... Elimina retorno a la *ROM.*
D5B5...RET .... Retorno al *BASIC.*

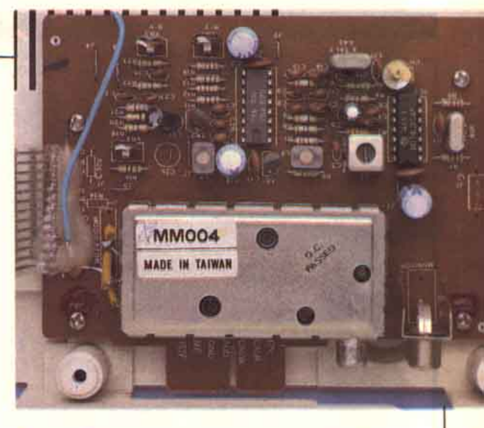


Foto 5. Interface del cassette y conexión para la antena.

Programa cargador

```
10 CLEAR 100,&HD54E
20 FOR A=&HD54E TO &HD5B5
30 READ A$: B=VAL("&H"+A$): C=C+B: POKE A,B: NEXT
40 IF C<>14257 THEN  CLS: PRINT"Error en las DATAS":
PRINT:LIST 70-190
50 DEFUSR=&HD54E
60 Z=USR(0):NEW
70 DATA e5,3e,c3,32,57,ff,21,78
80 DATA d5,22,58,ff,ed,73,5c,fe
90 DATA f3,3e,0f,d3,88,3e,dd,d3
100 DATA 8c,21,00,80,11,00,00,e5
110 DATA c1,ed,b0,3e,df,d3,8c,fb
120 DATA e1,c9,fe,c9,c0,ed,73,5c
130 DATA fe,dd,2a,03,fa,22,5e,fe
140 DATA f3,3e,0f,d3,88,3e,dd,d3
150 DATA 8c,21,00,80,11,00,00,1a
160 DATA ed,a0,2b,77,23,7c,b5,20
170 DATA f6,3e,df,d3,8c,fb,ed,7b
180 DATA 5c,fe,cd,50,37,dd,22,03
190 DATA fa,2a,5e,fe,23,7e,c1,c9
```

Manejo:

Salva el programa en cinta antes de ejecutarlo ya que una vez ejecutado se borra automáticamente si todo va bien.

Las rutinas en CM están en direcciones apropiadas para que el programa pueda usarse también con disco.

Cada vez que quieras conmutar los bancos, si no has apagado el ordenador después de correr el programa, usa el comando *SWITCH.*

Ejemplo: Ejecuta el programa anterior y a continuación escribe este otro:

```
10 SCREEN 2: COLOR 1,15
20 PRINT "BANCO 1"
30 SOUND 0,24
40 SOUND 7,254
50 SOUND 8,31
60 SOUND 12,1
70 SOUND 13,14
80 FOR A=1 TO 500: NEXT
90 SCREEN 0
100 SWITCH
110 GOTO 10
```

Tras teclear el programa teclea *RUN* y pulsa *ENTER*.

Cuando aparezca el mensaje O.K. podrás observar que si haces un *LIST*,. no te aparece el programa, esto ocurre porque el programa que has introducido ya está en el segundo banco.

Ahora introduce este otro programa:

```
10 PRINT "BANCO 2"
20 SOUND 0,208
30 SOUND 6,15
40 SOUND 8,30
50 SOUND 11,255
60 SOUND 13,8
70 FOR A=1 TO 500: NEXT
80 SWITCH
90 GOTO 10
```

Tecleando *RUN* y pulsando *ENTER* podrás observar cómo el ordenador ejecuta alternativamente los programas introducidos en ambos bancos.

Organización de la memoria RAM

Al encender el ordenador, la *RAM* se divide virtualmente en varias zonas. Cada una de estas zonas se destinará a usa función específica cuando manejamos el ordenador en *BASIC*.

La dirección en la que comienza o acaba cada una de estas zonas se denomina «dirección límite», y su valor se puede encontrar en las llamadas «variables del sistema».

La zona de variables del sistema es otra de las divisiones de la memoria *RAM*, y tiene la característica de ser la única zona de la memoria *RAM* cuya longitud no varía por necesidades del programa. Las direcciones límite entre las que se encuentra esta zona son la *F500 hex* y la *FFFF hex*, o sea, el final de la memoria *RAM*.

Las otras zonas en que se divide la memoria *RAM* varían de longitud a medida que el programa en *BASIC* va ocupando espacio en las mismas.

A continuación, y antes de ver para qué se utiliza cada una de las zonas que forman la memoria *RAM*, vamos a ver cuáles son y la forma de encontrar las direcciones límite de cada una de ellas. (Ver figuras 2 y 3).

Zonas de la memoria RAM

NOMBRE	DIRECCION LIMITE
	Inicio del programa BASIC.
Programa BASIC	— —
	Inicio de la tabla de variables.
Tabla de variables	— —
	Inicio de la tabla de matrices.
Tabla de matrices	— —
	Final de la tabla de matrices.
Area de trabajo	— —
	Puntero de la pila de BASIC.
Pila de BASIC	— —
	Final del espacio de cadenas.
Area de caenas	— — Puntero de cadenas.
	Inicio del área de cadenas.
Buffers de E/S	— —
	Tope de la memoria basic.
C. M. de usuario y sistema de disco	— —
	F500 hex.
Variables del sistema	— —
	FFFF hex.

Figura 2

Variables del sistema con direcciones límite

DIRECCION LIMITE	VARIABLE DEL S.	DIR. VARIABLE
Inico del programa BASIC	TXTTAB	F54A hex
Inicio de la tabla de variables	VARTAB	F7EE hex
Inicio de la tabla de matrices	ARYTAB	F7F0 hex
Final de la tabla de matrices	STREND	F7F2 hex
Puntero de la pila de BASIC	SAVSTK	F7DD hex
Final del área de cadenas	STKTOP	F546 hex
Puntero de cadenas	FRETOP	F7C7 hex
Inicio del espacio de cadenas	MEMSIZ	F7A2 hex

Para hallar las direcciones en que empiezan y acaban las zonas de la figura 2, hemos de leer *(PEEK)* en las variables del sistema de la figura 3 el contenido de estas.

Hemos de tener en cuenta que la dirección buscada será un número comprendido entre 0 y 65.535, ó FFFF en hexadecimal, por lo que la dirección buscada estará almacenada en dos octetos de memoria.

El resultado lo podemos obtener en formato decimal o en formato hexadecimal aplicando las fórmulas siguientes:

Dirección buscada = *PEEK* (variable del sistema) + 256 * *PEEK* (variable del sistema + 1) resultado en decimal.

O bien:

Dirección buscada = *HEX$* (*PEEK* (variable del sistema) + 256 * *PEEK* (variable del sistema + 1)) resultado en hexadecimal.

Venerando Solis

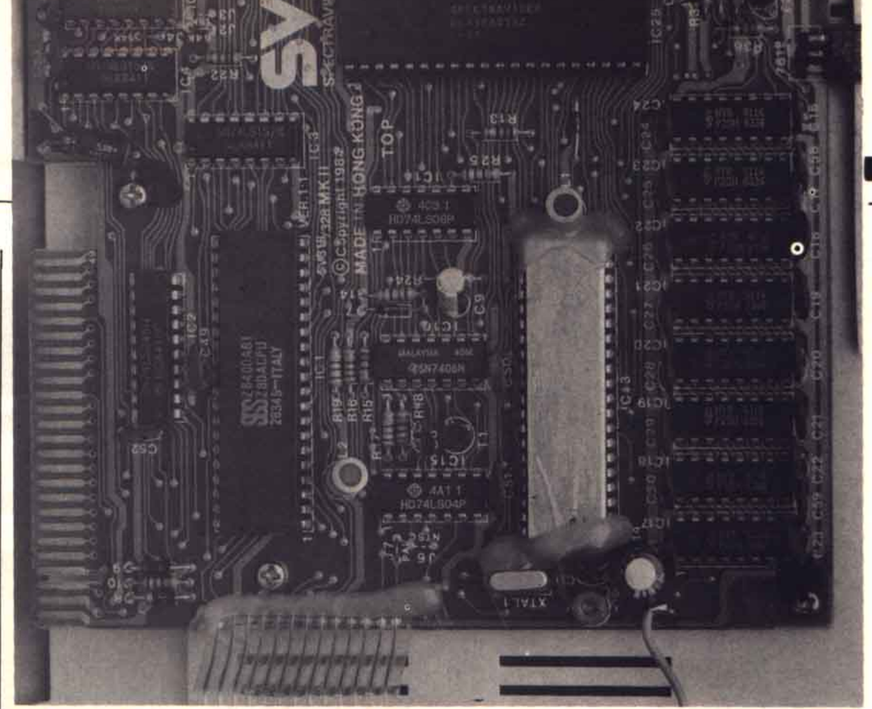


Foto 6. Interface para cualquiera de las unidades de expansión que admite.

Toshiba HX-20

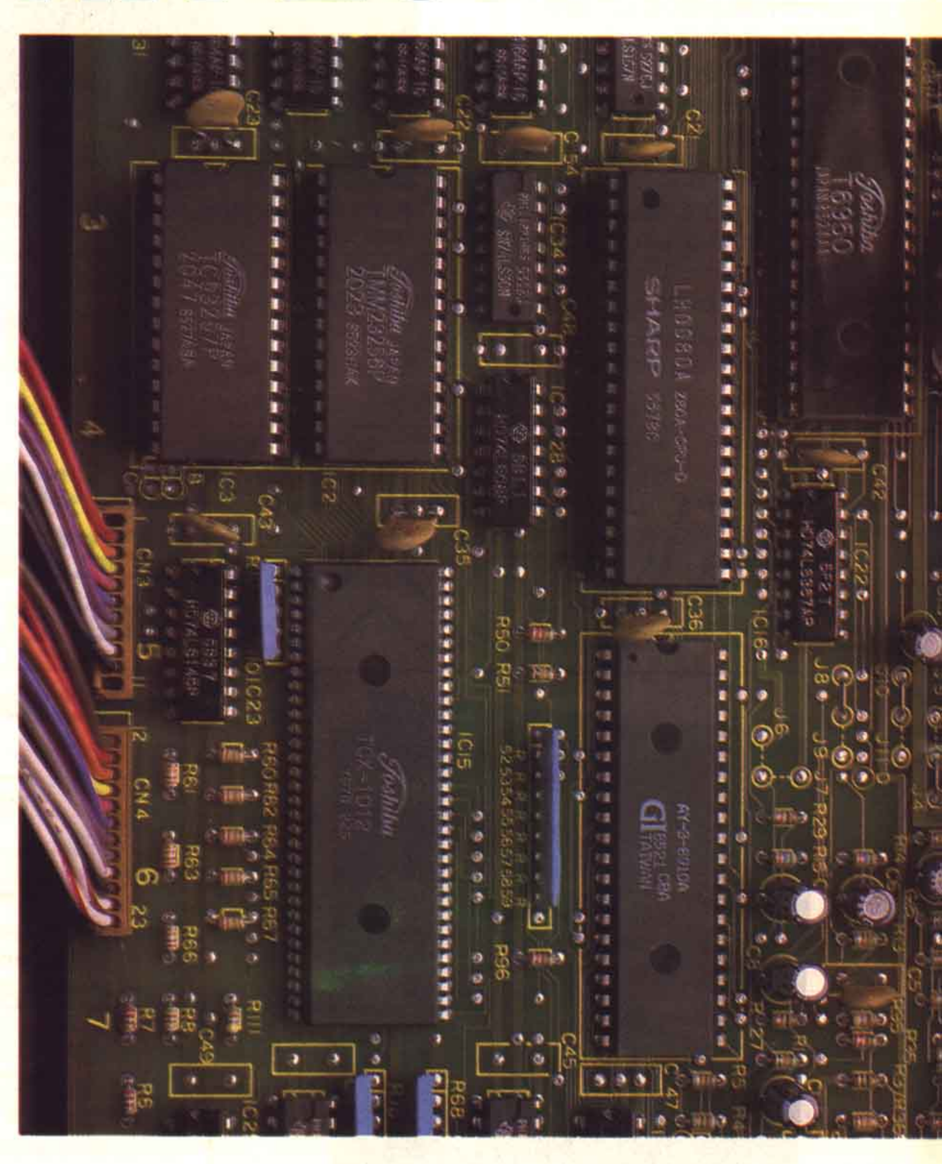
En la actualidad, muchos usuarios esperan impacientes la aparición de la segunda generación de MSX, la cual se está haciendo de rogar. Han aparecido ordenadores basados en los primeros modelos, pero con algunas características que los destacan del resto de los existentes, tal es el caso del Toshiba HX-20 y del Spectravídeo X'press. Sin embargo, este mes comentaremos únicamente el primero de ellos.

El Toshiba HX-20 (y su modelo mejorado el HX-22) están situados en un escalafón intermedio entre la 1.ª y 2.ª generación de MSX. Su principal cualidad es la de poseer además de un potente tratamiento de textos, un disco *RAM,* pero esto lo veremos más adelante. Por el momento describiremos el ordenador en si, para luego entrar en los detalles que lo destacan.

El aparato, externamente, posee un diseño muy atractivo. El teclado incluye todas las teclas del estándar, ofreciendo a su vez un tacto agradable, aunque al principio resulta un tanto extraño. Pero a partir de aquí empezamos a notar las primeras diferencias con respecto a un MSX normal.

La fiesta se inicia al conectar el ordenador. Primero aparece el sempiterno logotipo que identifica el sistema MSX, y posteriormente una pantalla de presentación muy llamativa, que nos muestra un conocido dibujo de los edificios de Manhattan. Una vez repuestos, pulsamos una tecla y nos topamos con un menú de dos opciones: *BASIC* y tratamiento de Textos.

El Tratamiento de Textos que lleva incorporado en *ROM* es una utilidad a la que no hemos encontrado pega alguna, con la salvedad de que hay que acostumbrarse (como es lógico).

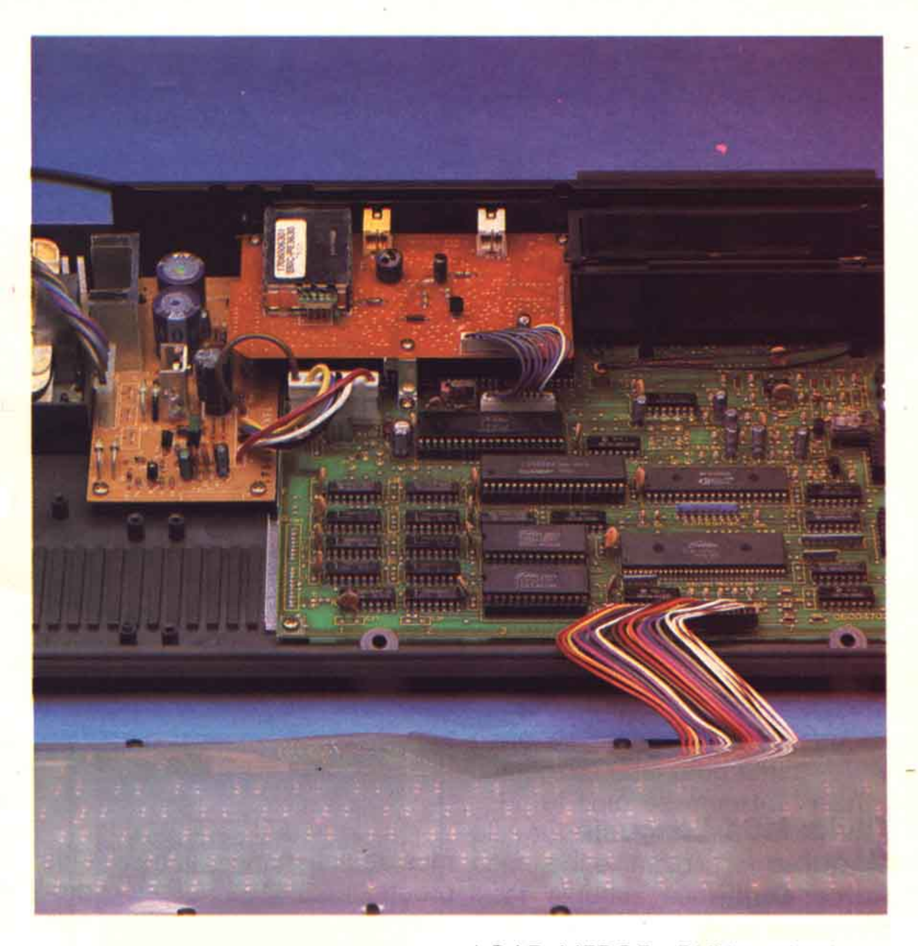


Tiene unos pequeños puntos conflictivos, pero nada que el completo manual que acompaña el aparato no pueda resolver. Sin embargo, cabe destacar que viene preparado para comunicarse vía *RS-232C,* algo que todavía no habíamos visto hasta la fecha (hay que resaltar que el HX-22 tiene este interface incorporado y que lo comentaremos en números sucesivos) y que redundará en benefi-

cio de otros ordenadores dentro de la norma MSX, puesto que es totalmente viable mandar información de un ordenador a otro a través de dicho interface.

Por otro lado, el *BASIC* resulta algo más potente, gracias a la incorporación de instrucciones que se utilizan con el disco *RAM* que lleva este ordenador. Dicho disco, es un espacio reservado en la memoria del aparato que se maneja de la misma forma en que lo haríamos con un disco normal, con la salvedad de que no admite la creación y acceso a ficheros aleatorios. Entre las instrucciones nuevas que aporta este *BASIC* podemos destacar las siguientes:

—CALL MEMINI; inicializa la función del disco *RAM.*





—CALL MEMOFF; anula la función del disco *RAM,* no permitiendo otro comando que acceda al disco y borrando todos los ficheros que tengamos en ese momento.

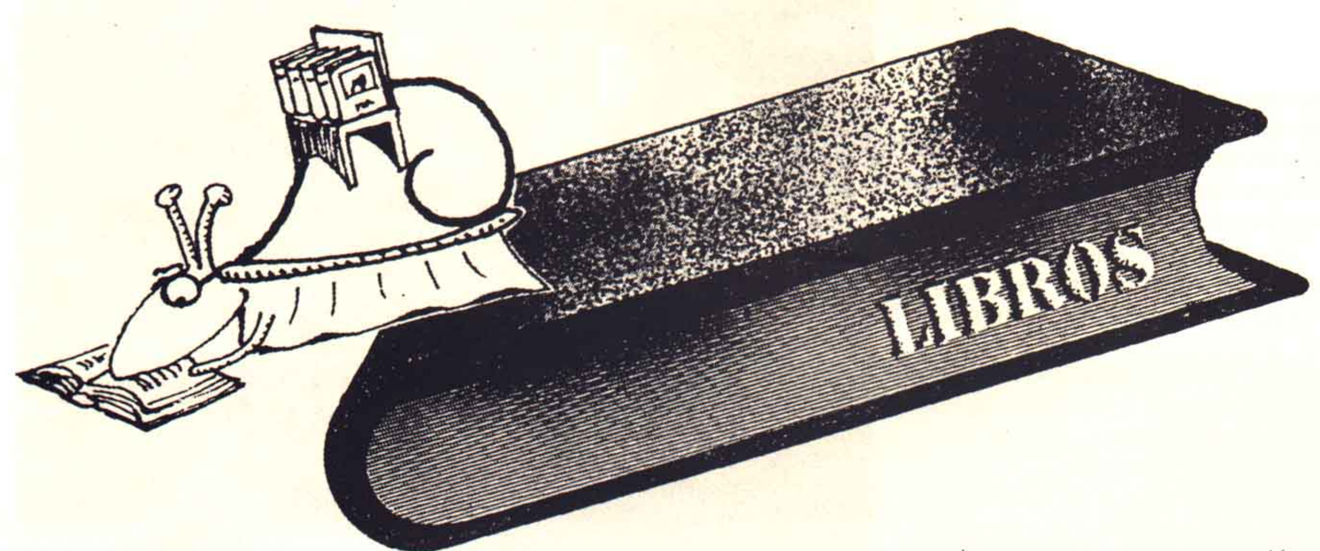
—CALL MFILES; muestra el directorio del disco *RAM.*

—CALL MKILL; borra un fichero del disco.

Para salvar o cargar ficheros, se utilizan las instrucciones. *SAVE, LOAD, MERGE* y *RUN* con la sintaxis habitual especificando *MEM* como nomitivo.

Estas son sólo algunas de las más importantes. El resto de las instrucciones son las mismas que tienen todos los ordenadores del estándar, por lo que no afectan a la compatibilidad del sistema, aunque los programas que utilicen las instrucciones del *BASIC* extendido del Toshiba HX-20, no funcionarán en otros ordenadores.

Los manuales que acompañan al ordenador son muy completos y están todos en castellano. El primero es una guía del usuario que explica las conexiones del aparato, así como los primeros pasos a seguir con el ordenador. El segundo manual es una guía de referencias donde se explican todos los comandos y funciones del *BASIC MSX,* mientras que el último corresponde a las instrucciones del procesador de textos y se dedica a explicar el *BASIC* extendido de este modelo. Es, sin lugar a dudas, una buena opción



Título: MSX Lenguaje Máquina
Autor: Dullin - Strassenburg
Editorial: Ferre Moret S.A.
Páginas: 305

Nosotros los programadores, o aquellos que nos iniciamos en el mundo de la informática, una de nuestras grandes ilusiones es el crear un programa por nosotros mismos, pero que sea de utilidad, y muchas veces no contamos con la suficiente teoría o práctica para conseguirlo, por ello nos es necesario recurrir a ejemplos ya resueltos que nos informen sobre el tema que deseemos tratar. Refiriéndonos al estándar, *102 Programas para MSX* es uno de los manuales indispensables para conseguir una gran ayuda y experiencia en

Aunque el libro en su mayor parte tiene un carácter didáctico, es útil a su vez para aumentar nuestra colección de software, con ejemplos únicos y fáciles, hasta para los más pequeños.
todo lo que se refiere a la creación de programas para ordenador.

No obstante, aunque nos refiramos de una manera general a los ordenadores MSX, los cien programas que se encuentran en este libro pueden ser utilizados en todos los ordenadores que utilizan el *BASIC* Microsoft.



La brevedad de los programas propuestos, permitirá teclearlos fácilmente sin error y servirán de base para programas de mayor envergadura.

La organización de la obra ha sido estimada en su mayor parte para lograr una mayor expresión de las técnicas de programación actuales, por ello comienza con una pequeña introducción sobre los más conocidos comandos y elementos de programación del *BASIC*.

La disposición del libro, nos permite clasificar los programas de 1 a 5, en orden creciente de complejidad. Un primer nivel, en que las instrucciones elementales del lenguaje *BASIC*, como *LET*, *PRINT*, etc., nos llevará. recordar un poco aquellos elementos esenciales que habíamos olvidado.

Un segundo nivel en que la utilización de *DATAS*, *READ*, O LA FORMA DE DIMENSIONAR la pantalla serán los temas que nos preocupen.

Y como síntesis nos referiremos al nivel 5, en el que la utilización de *VPEEK* y *POKE*, nos será de gran ayuda.

La presentación de los juegos es muy sistemática, comienza con una corta descripción y un estudio de las líneas más importantes. A continuación se propone un ejemplo de los programas y su ejecución.

En resumen, podríamos decir, que el objetivo de este libro es aprender distrayéndose. A lo largo de estos 102 programas, se le

guiará en la exploración del *BASIC,* y prácticamente todas las posibilidades de este lenguaje se examinarán en este libro.

Como ejemplos prácticos de mayor interés para el lector-programador encontramos la búsqueda de un elemento en la cadena *'MID$,* así, si *A$="BUENOS DIAS",MID$(A$,3,1)* proporciona el carácter *E.*

— Manejo de cadenas de alta resolución y gráficos. Utilizando *LEFT$ (A$,2)* permite extraer los primeros caracteres de la izquierda de la palabra *A$,* y viceversa para *RIGHT$.*

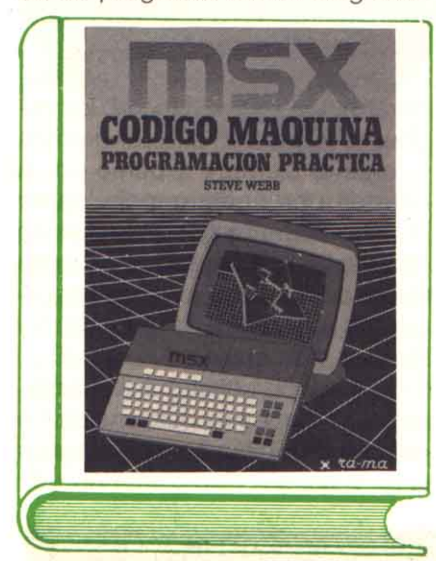
— Y como última nota, mencionar la multitud de juegos que le pueden ayudar a entender mejor el lenguaje *BASIC* de programación que le ayudarán a saber más divirtiéndose al mismo tiempo, como es el caso de: Ruleta, Jackpot, Tablero o Solfeo.

Título: Código Máquina.
Programación Básica
Autor: Steve Webb
Editorial: Ra-Ma
Páginas: 127

¿Qué es Código Máquina? Es un lenguaje de programación, pero no resulta tan complejo como probablemente te hayan inducido a creer. La intención de este libro es introducirte en la programación en Código Máquina de los ordenadores MSX, por lo que no se refiere a este tipo de programación de una forma general.

Si tienes conocimientos del *BASIC* te será muy fácil captar los conceptos sobre programación en Código Máquina que a lo largo de este libro se dan, pero no pienses que va a resultar sencillo.

La disposición del libro en capítulos, permite asimilar con facilidad todas las enseñanzas que vas a recibir, pero antes de pasar de uno a otro debes haberlos entendido perfectamente, ya que los primeros capítulos del libro encierran la teoría, respecto del almacenamiento de los números, y las equivalencias con el *BASIC.* Haciendo referencia al capítulo principal, podemos señalar, como nos describe un programa en un diagrama



de flujo, y luego transcribir los bloques de ese diagrama, como cortas y simples rutinas o subrutinas.

En los últimos capítulos encontraremos cómo sacar provecho de las avanzadas facilidades del MSX, tales como las figuras móviles y definibles por el usuario, y su sonido. Si deseas obtener el máximo beneficio de estas facilidades, necesitarás dominar el Código Máquina.

Una de las mejores técnicas de enseñanza es comenzar desde cero, pero con una cierta base desde luego, que en este caso se la podríamos atribuir al lenguaje *BASIC,* y es en esta técnica en la que se basa nuestro libro. No es un tratado científico en el que se descubra todo el entrañable lenguaje de Código Máquina pero, desde el primer momento, asienta unas bases claras y definitivas.

Pero verdaderamente ¿qué es el Código Máquina? Es, en una palabra, el lenguaje subsidiario del *BASIC,* pero con la característica de que su empleo en los programas facilita la creación de los mismos y su ejecución, así como la sintetización de varias rutinas en una, ahorrando con ello una gran parte de tiempo y facilitando la compresión del programa.

En el libro se muestran ejemplos sobre el tema que se explica, como por ejemplo la forma de enlazar algunas sentencias como *PRINT, GOSUB, IF,* etc., con un juego «Invasor del Espacio», el cual te ayudará a reconocer tus errores, y al mismo tiempo nos presenta una serie de pruebas auto-evaluativas de los conocimientos adquiridos a lo largo del libro.

Esto es en síntesis el contenido del libro, aunque nos hemos dejado numerosos aspectos sin destacar, sin embargo debemos hacer mención de que aunque lo importante de un libro es su contenido — sobre todo en un libro didáctico— también es muy importante la forma de presentarlo, el tamaño de las letras, las expresiones que utiliza, e incluso el papel que emplea debe ser en todo momento acorde con el tema que trata, y en estos aspectos creemos que se podría superar, así como las erratas en algunas instrucciones (que son pocas), pero que desmerecen la calidad del ejemplar.

Sin más comentarios al respecto, y dejando claro que el contenido del libro merece ser leído, os dejamos a vuestro arbitrio las conclusiones finales.

SOFTWARE

Programa: Turmoil
Tipo: Juego
Distribuidor: Mind Games
Formato: Cassette

Una cosa es hacer un juego y otra hacer un juego de software. *Turmoil* es un verdadero juego de software tanto en contenido como en gráficos, como en sonido. Muchas veces encontramos juegos que nos parecen malos o que no se ven bien en el ordenador. La mayoría de las veces resulta que lo que estamos viendo son juegos y no juegos de software. Pero esto no le ocurrirá cuando comience a jugar en nuestro *Turmoil.*

Imaginemos un escenario creado en la cuna del petróleo, elemento base de nuestro juego, es decir, Arabia, los países del Lejano Oriente, etc.

Y como motivo central, los pozos petrolíferos del Gran Sultán que gobierna nuestro sultanato imaginario.

Una vez conocido el enclave de nuestra aventura, haremos mención a nuestro personaje, en una palabra a nosotros, pero eso lo dejaremos para más tarde, así te invitamos a sumergirte en esta fabulosa aventura.

Figurémonos una inmensa nave de pozos de petróleo, guardada por los más temibles tuaregs del desierto, cuya habilidad en las artes de la espada y los cuchillos serán los enemigos de nuestra aventura. Pero ya es hora de que expliquemos en qué consiste esta sin igual aventura.

Seremos un ingeniero que cuidará del perfecto estado de los coches del Sultán, que tendremos que tener siempre a punto por si ocurre una emergencia, pero nuestro acuerdo con el Sultán no ha sido tan del agrado suyo como nuestro, y la única forma de escapar es un tanto peligrosa para nuestra vida, pero es la única.

Se trata de conseguir llenar todos los tanques de gasolina de los coches del Sultán para que no nos puedan perseguir, pero no sabemos dónde están los coches y la vigilancia es perfecta, tendremos que utilizar una gran habilidad para no encontrarnos con los árabes que gobiernan el recinto.

Por tanto, debemos ser muy rápidos a la hora de encontrar los surtidores y llegar a los depósitos de los coches en el menor tiempo posible, ya que desde el lugar donde se nos surte de gasolina, hasta los vehículos, tendremos que luchar contra los árabes, o bien despistarles ya que son muy pocas las armas con las que contamos. Nuestra astucia, velocidad, y la propia gasolina con la que podremos hacer que se escurran y desaparezcan por un rato, constituyen nuestras mejores armas.

Hemos mencionado los alicientes del juego más notables, pero la sensación que experimentarás será mucho mayor que la que ahora mencionamos.

Hemos hablado antes de lo que son verdaderos juegos de software, por tanto es necesario que hagamos referencia a sus gráficos, sonido, etc.

Respecto a la creación de los sprites de que consta el juego, son de una técnica esmerada y de una concienzuda realización, podemos destacar la figura de los árabes y su caricatura, así como la perfecta realización de los coches, así como el rugir de los motores cuando conseguimos escapar con el.

¿Qué le parece experimentar una aventura en los extraños mundos del petróleo y de los secretos? La respuesta es *Turmoil.*



Puntuación:
Presentación: 8
Claridad: 8
Rapidez: 9
Adicción: 9

Programa: Chukie Egg
Tipo: Juego
Distribuidor: Microbyte
Formato: Cassette

Estamos acostumbrados a ver un tipo de programas que tienen como base la guerra o la defensa de la tierra, pero hay otros muchos que tratan de temas que nos resultan más acordes con la realidad de la vida. Este es el caso de la convivencia en las granjas, el contacto con la Naturaleza, o los simples quehaceres de la vida doméstica y cotidiana.

Porder pasar un rato entretenido vigilando nuestra granja es uno de los entretenimientos que este juego nos proporciona, aunque esto nos infunda una cierta monotonía, comprobará mientras se desarrolle el juego que no es cierto y que también nos ofrecerá multitud de alicientes.

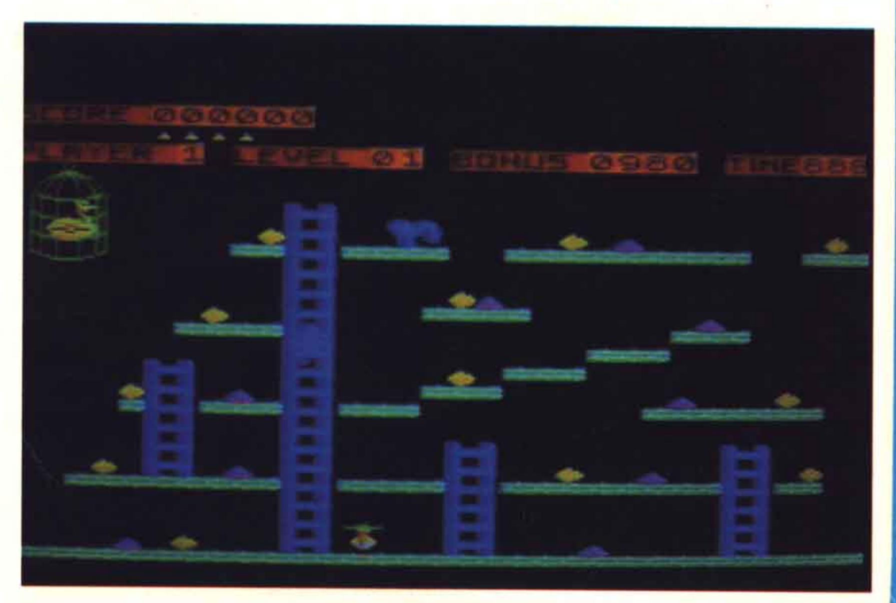
Chukie Egg, se desarrolla en una granja avícola, un poco desordenada, y usted debe vigilar que este desorden no le produzca pérdidas en sus almacenes. El grano, por una equivocación de los almacenadores, ha sido esparcido por todos los rincones de la granja y nuestros animalitos han visto hechos realidad sus sueños al encontrarse la comida por todos los rincones.

Nuestra tarea consiste en recogerlo antes que ellos lo encuentren. Pero la disposición de la granja es un tanto complicada y los pollitos andan sueltos por todas partes. Al principio tan solo algunos se han dado cuenta, pero llegará un momento que será imposible recogerlo, por lo que debemos actuar rápidamente y recoger y vigilar todos los departamentos o pisos de los que consta la granja.

Pero este juego no sólo encierra rapidez y control, debemos tener mucho cuidado, porque a medida que los pollos comen, se van haciendo más y más fuertes, y cada vez que tropecemos con uno de ellos perderemos una vida. No obstante, si logra recoger la mayor parte de los granos conseguirá más puntos, pero el juego se empezará a complicar y tendrá que desarrollar una mayor velocidad a la hora de recogerlo.

Pero la diversión del juego no sólo se encuentra aquí, sino que los gráficos, sencillos pero bien realizados, hacen que podamos visionar de una forma clara los pollos, los granos, y el recinto de la granja en que nos encontramos.

El juego consta de tres vidas,



SOFTWARE

con las que partirá desde el primer momento, pero tenga cuidado en conservarlas ya que sino se da prisa, se verá arruinado, y lo que es peor, el juego se terminará.

Chukie Egg es un juego divertido y entretenido hasta para los más pequeños. Fácil de jugar y sin grandes complicaciones.

Puntuación:
Presentación: 7
Claridad: 7
Rapidez: 7
Adicción: 6

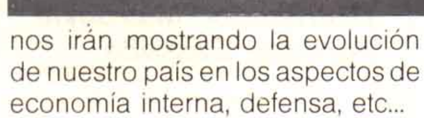
Programa: Juego de la Moncloa
Tipo: Juego
Distribuidor: A.B.C.
Formato: Cassette

¿Qué le parece por unos momentos sentirse presidente de una nación, en este caso España?, un poco difícil, ¿verdad? Este es un juego de estrategia que tiene como base los aconteceres políticos vistos desde la pura realidad, con sus victorias y sus derrotas. En estos momentos políticos que estamos viviendo, nada mejor que poder pasar un rato divertido sin pensar en las repercusiones que originarían medidas de política gubernamental.

Este juego nos hará familiarizarnos con todas las tareas y problemas que encierra un país.

En el aspecto técnico del programa (juego), diremos que su elaboración en cuanto a contenido, que más tarde comentaremos, es bastante completa, pero su creación, es decir, el lenguaje de programación utilizado, BASIC, le hace un poco lento a la hora de dar los resultados, aunque esto no encierra mayor problema, ya que nos permite tomar las decisiones con más tranquilidad y reflexión. No hay mal que por bien no venga.

El programa se basa en una serie de gráficos que, situados en la parte superioir de nuestro monitor, nos irán mostrando la evolución de nuestro país en los aspectos de economía interna, defensa, etc...



Podremos consultar con nuestros ministros si vemos que no podemos solucionar por nosotros mismos los problemas que conlleva gobernar un país, pero mucho cuidado, porque las medidas que adoptemos serán bajo nuestra resposabilidad, y si no son las acertadas recibiremos la dimisión de alguno de nuestros ministros, creando así un conflicto más.

Contaremos a su vez con la colaboración de un secretario que nos tendrá al día de todas las noticias que sucedan durante nuestros cien días de mandato. También contaremos con la ayuda de los medios informativos, que nos harán llegar las noticias con la mayor facilidad, ya que las catástrofes y las inundaciones también repercuten en la economía de un país.

Quizás donde debemos prestar mayor atención es en los gráficos iniciales, paro e inflación, principales problemas de un país.

No obstante, podremos cambiar de gráficos si queremos ver la evolución de otros aspectos, como pueden ser población activa, producto interior bruto, etc.

Y un elemento que sin duda será de gran ayuda: los sondeos, que nos dirán el grado de optimismo o pesimismo de los sectores que componen la población, el pueblo, la política, los banqueros o los empresarios.

Si durante nuestros cien días de mandato los índices no aumentan alarmantemente y todo transcurre dentro de una normalidad, podre-



mos seguir gobernando, pero llegar a la conjunción de todos los intereses es muy difícil y en cualquier momento corremos el peligro de una rebelión o de un golpe de Estado.

En este juego podremos ver de cerca las repercusiones del IVA de una manera simbólica y cuáles son sus repercusiones, así como el aumento de salarios o la simple disminución de la jornada laboral.

Si la política ha sido una de sus frustraciones, ahora tiene tiempo de hacerlas realidad, y quizas poco a poco, pueda ser un día presidente de gobierno.

Puntuación:
Presentación: 7
Claridad: 7
Rapidez: 7
Adicción: 9

Programa: Dominó
Tipo: Juego
Distribuidor: Idealogic
Formato: Cassette

La imaginación del hombre es infinita, pero tiene momentos en que se agota y sufre un estancamiento, por lo que muchas veces decide volver atrás y empezar de nuevo.

Esto nos sirve de introducción para darnos cuenta de que la mayoría de los juegos que se encuentran en el mercado se pueden agrupar bajo unas características comunes en cuatro o cinco grupos. Esto es debido a que la mente del hombre comenzará dentro de poco tiempo a sufrir una recesión y existirá un estancamiento en su forma de crear e imaginar.

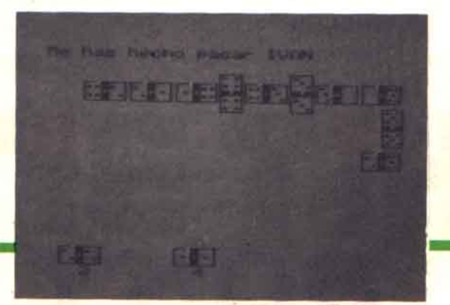
Pero lo bueno es darse cuenta de que esto ocurrirá y comenzar ahora a recopilar datos que necesitemos para darle un respiro a nuestra imaginación. Este es el caso de *Dominó* que ha surgido en un momento clave en el mercado de los juegos, inundado por guerras, aventuras o simples pasatiempos.

Pero *Dominó* es un juego que no necesita presentación porque de todos es conocido y recordado.

Las reglas del *Dominó* son fáciles de entender. El juego consta de 28 fichas, cada una con un valor. Hay que repartir un número de ellas, que en este caso será de siete, puesto que jugará el ordenador contra nosotros. Una vez repartidas las fichas comenzaremos a jugar, pero antes de comenzar tendremos que trazar una línea de juego, para que si se termina la partida no nos quedemos con un número elevado de puntos.

Este es el desarrollo esencial del juego, pero vamos a destacar otros aspectos que aún no hemos mencionado. Comenzaremos destacando la invitación a jugar con que empieza el juego, preguntándonos nuestro nombre. De esta forma se dirigirá a nosotros durante la partida. Podemos elegir el número de partidas que quertamos jugar, pudiendo elegir de uno a veinte.

La técnica que desarrolla nuestro adversario es lógica y sofisticada, por lo que tendremos que pensar mucho antes de posicionar una ficha.



Las características del juego son de fácil entendimiento, ya que se irán clasificando cada uno de los pasos que debemos seguir, mientras que la presentación no tiene mucho que destacar, pero se ha realizado de acuerdo con las necesidades del juego, por lo que será muy fácil ver los puntos y las fichas que entran en el juego.

La forma de colocarlas es muy sencilla. Tan sólo con dar el número de ficha que ocupe un lugar del 1 al siete, y eso será todo, al final la suma de los resultados será dada por el ordenador y él mismo nos dirá si hemos ganado o no.

Pero no descuidemos nuestro juego si no queremos vernos obligados a pasar en el resto del juego.

Puntuación:
Presentación: 7
Claridad: 9
Rapidez: 8
Adicción: 9

Programa: Star Force
Tipo: Juego
Distribuidor: S.E.R.M.A.
Formato: BEE CARD

La era del espacio ha llegado de una forma inminente hacia nosotros desde la literatura, el cine y la informática.

Las más sutiles técnicas de combate hoy en día se asemejan de una manera exacta a las más sofisticadas armas de la Guerra de las Galaxias o a esas fantásticas naves que surcan el espacio.

Programas Sony MSX, para lo que guste ordenar.



JUEGOS



GESTION



EDUCATIVOS



APLICACIONES





SOFTWARE

PROGRAMAS SONY MSX

Educativos

- Monkey Academy
- Alfamat
- Viaje Espacial
- Multipuzzle
- Noria de Números
- Corro de Formas
- Coconuts
- Yo Calculo
- Selva de Letras
- El Cubo Informático
- Electro-graf
- El Rancho
- Teclas Divertidas
- Boing Boning
- Compulandia
- Mil Caras
- Logo
- Países Mundo-1
- Países Mundo-2
- Tutor
- Computador Adivino
- Aprend. Inglés-1
- Aprend. Inglés-2
- Cosmos
- Curso de Básic
- Juego de Números

Juegos

- Antártic Adventure
- Athletic Land
- Sparkie
- Juno First
- Car Jamboree
- Battle Cross
- Crazy Train
- Mouser
- Computer Billiards
- Alí Babá
- Track & Field-I
- Track & Field-II
- Dorodon
- Chess (Ajedrez)
- Senjo
- E.I.
- Lode Runner
- Super Tennis
- Backgammon
- Super Golf
- Hustler
- Binary Land
- Driller Tanks
- Stop the Express
- Ninja
- Les Flics
- La Pulga
- The Snowman
- Cubit
- Pack 16K
- Fútbol
- Kung Fu
- Batalla Tanques
- Mr. Wong
- Xixolog
- Buggy
- Sweet Acorn
- Peetan
- Jump Coaster
- Buggy 84
- 3D Water Driver
- Pinky Chase
- Wedding Bells
- Fightting Rider

Aplicación

- Memoria Ram 4 K
- Creative Greetings
- Character Collect
- Quinielas y Reducciones
- Pascal
- Ensamblador
- Generador Juegos

Gestión

- Hoja de Cálculo
- Homewriter
- Control Stocks
- Contabilidad Personal
- Ficheros
- Procesador de Textos
- Control Stocks
- Vencimientos
- Contabilidad 1.500



Nos encontramos en el año 2.010, y el espacio ha sido envadido por un gran número de naves flotantes que atacan y destruyen cualquier objeto que encuentren a su paso. Estas naves reciben el nombre de *Gordus.*

La unión de todas las fuerzas del espacio que luchan contra el poder de estas naves nodrizas, de ataque mortífero recibe el nombre de *Alianza.* Pero no nos vamos a detener en narrar los motivos de este dominio de una forma exacta, le dejaremos que sea usted quien se dé cuenta de la situación.

Tu misión, en síntexis, consiste en destruir todo enemigo que encuentres a tu paso, ¿que cómo los conocerás?, espera que lleguen...

Las características del juego hace que por sus gráficos y por los elementos que lo constituyen, sea muy interesante. Las pantallas, 24, el sonido de lucha entre la vida y la muerte y la rapidez y agilidad del programa hacen que te metas dentro del problema del Universo y te sientas indispensable.

La estrategia que debes seguir será la de no quedarte parado nunca y efectuar un ataque continuo, así irás sumando puntos pero, además de destruir a tus enemigos tendrás que fijarte también en el terreno, puesto que será ahí donde consigas el mayor número de puntos y donde, a su vez, se encontrarán los enemigos más peligrosos.

La mayor parte del tiempo volarás sobre un largo camino de naves nodrizas que siguen un destino, por lo que no debes de dejar de observar tu nave, pues en el fondo de la pantalla no se diferencia apenas el color y muchas veces no sabrás tu posición debido al constante movimiento que debes mantener.

Pero este no es el peligro más importante, tu nave se encontrará siempre en movimiento, aunque no estés controlándola en ese momento. Ciertos enemigos necesitarán un disparo para ser visibles y cuatro para sucumbir, logrando así 2000 puntos.

Los *Zmudastegui,* el tipo de nave más peligrosa, se encontrará siempre en los laterales de las na-



ves nodriza y su aniquilación nos dara 80000 puntos.

Pero. ¡sorpresa!, existe un enemigo escondido con el que obtendremos un millón de puntos, aunque no será fácil conseguirlos.

Puntuación:
Presentación: 9
Claridad: 8
Rapidez: 9
Adicción: 8



Formulación química

El programa que a continuación se detalla, presenta la solución al problema que puede suponer para los estudiantes la formulación química. Como el programa es susceptible a mejoras, a continuación se presenta un esquema del mismo.

n.º línea	
10 a 210	Presentación e instrucciones.
220 a 300	Comprueba si es óxido, peróxido, hidróxido, ácido o sal.
350 a 760	Tratamiento de óxidos y peróxidos dependiendo de la terminación «OSO» o «ICO».
380 a 690	Da valencias códigos a los elementos que puedan formar óxidos, peróxidos e hidróxidos.
770 a 780	Tratamiento de hidróxidos.
790 a 870	Tratamiento de anhídridos.
880 a1130	Tratamiento de ácidos mirando los códigos «KK» de los diferentes grupos.
1130 a 1340	Tratamiento de los compuestos del Boro.
1350 a 1540	Da valencias y códigos a los elementos que puedan formar anhídridos y ácidos.
1550 a 1580	Convierte datos numéricos en alfanuméricos para su correcta distribución en la fórmula final.

Variables

— En óxidos, peróxidos e hidróxidos.

V1 = Menor valencia con que actúa.
V2 = Mayor valencia con que actúa.
Q$ = Tres primeras letras del elemento.
CO$ = Denominación simplificada.

— En anhídridos y ácidos.

V1 = Menor valencia en elementos que tengan cuatro.
V2 = Menor valencia en elementos que tengan dos, y segunda valencia en los de cuatro.
V3 = Mayor valencia en los elementos que tengan dos, y tercera valencia en los de cuatro.
V4 = Mayor valencia en elementos que tengan cuatro.
M$ = Tres primeras letras del elemento.
C$ = Denominación simplificada.
KK = 1 en elementos que formen compuestos como el F1, C1, Br, I.
KK = 2 en elementos que formen compuestos como el S, Se, Te.
KK = 3 en elementos que formen compuestos como el P, As, Sb.
KK = 4 en elementos que formen compuestos como el C, Si.
KK = 5 en el caso especial del Boro.

Fernando Pombar
Guipuzcoa

```
10 REM ##FORMULACION POR F.POMBAR#
   #
20 REM ##PRESENTACION E INSTRUCCIO
   NES##
30 KEYOFF:COLOR 1,15,15:CLS:LOCATE
    10,1:PRINT"**FORMULACION**"
40 LOCATE 3,5:PRINT"¿Quieres instr
   ucciones (S/N)?"
50 I$=INKEY$:IF I$="S" OR I$="s" T
   HEN 80
60 IF I$="" THEN 50
70 GOTO 230
80 CLS:LOCATE 10,3:PRINT"-INSTRUCC
   IONES-"
90 LOCATE 3,6:PRINT"Este programa
   formula gran parte  de los óx
   idos,peróxidos,anhídridos,hid
   róxidos,ácidos y sales en la
   forma  TRADICIONAL.Para ello,
    deberás escri-bir el nombre
   del compuesto,sin utilizar la
    preposición -DE-."
100 LOCATE 3,13:PRINT"Por ejemplo,
    el OXIDO DE LITIO,deberás esc
    ribirlo:OXIDO LITICO,de lo co
    ntrario,el ordenador no te lo
     entende-rá."
110 LOCATE 3,20:PRINT"PARA CONTINU
    AR PULSA UNA TECLA"
120 IF INKEY$=""THEN 120 ELSE 130
130 CLS:LOCATE 3,5:PRINT"Los pasos
     a seguir para las sales serí
    an los siguientes por ejemplo
     en el PERMANGANATO POTASICO:
140 LOCATE 3,9:PRINT"1.¿NOMBRE DEL
     COMPUESTO? SAL + Re turn"
150 LOCATE 3,12:PRINT"2.¿1a COMPON
    ENTE? PERMANGANATO +  Return"
160 LOCATE 3,15:PRINT"3.¿2a COMPON
    ENTE? POTASICO + Re - turn"
170 LOCATE 3,20:PRINT"PARA CONTINU
    AR PULSA UNA TECLA"
180 IF INKEY$=""THEN 180 ELSE 190
190 CLS:LOCATE 3,5:PRINT"Esperamos
     que este programa te ayude e
    n tu formulación.¡Ah,otra cos
    a!,alo mejor algunos compuest
    os el ordenador no los compre
    nde,lo cual no quie-re decir
    que no existan.Esperamos ques
    ean pocos."
200 LOCATE 3,20:PRINT"PARA COMENZA
    R PULSA UNA TECLA"
210 IF INKEY$=""THEN 210 ELSE 230
220 REM ##DEFINICION DE VARIABLES#
    #
230 COLOR 15,6,6:CLEAR:CLS:INPUT"¿
    NOMBRE DEL COMPUESTO";N$
240 L$=LEFT$(N$,2):D$=RIGHT$(N$,3)
250 IF L$="OX"THEN 360
260 IF L$="HI"THEN 780
270 IF L$="AN"THEN 800
280 IF L$="AC"THEN 890
290 IF L$="SA"THEN 1130
300 IF L$="PE"AND MID$(N$,4,2)="OX
    "THEN PE=1:GOTO 360
310 LOCATE 10,13:PRINT"ESE NO LO C
    ONOZCO"
320 LOCATE 6,12:PRINT"┌──────
    ───────────────────┐":LOCATE 6
    ,13:PRINT"│ ":LOCATE 29,13
    :PRINT"│":LOCATE 6,14:PRINT
    "└────────────────────────┘":
    LOCATE 7,21:PRINT"────────
    ──────────"
330 LOCATE 7,20:PRINT"PULSA UNA TE
    CLA (Z=END)":I$=INKEY$:IF I$=
    "Z" THEN BEEP:CLS:COLOR 15,4,
    4:KEYON:END ELSE IF I$=""THEN
     330
340 BEEP:GOTO 230
350 REM ##TRATAMIENTO DE OXIDOS Y
    PEROXIDOS##
360 O$=" O":X=2
370 IF PE=1 THEN Q$=MID$(N$,10,3)
    ELSE Q$=MID$(N$,7,3)
380 REM ##VALENCIAS DE LOS METALES
    ##
390 IF Q$="LIT"THEN CO$=" Li":V2=1
400 IF Q$="SOD"THEN CO$=" Na":V2=1
410 IF Q$="POT"THEN CO$=" K":V2=1
420 IF Q$="RUB"THEN CO$=" Rb":V2=1
430 IF Q$="CES"THEN CO$=" Cs":V2=1
440 IF Q$="AMO"THEN CO$=" (NH4)":V
    2=1
450 IF Q$="ARG"THEN CO$=" Ag":V2=1
460 IF  Q$="FRA"THEN CO$=" Fr":V2=
    1
470 IF Q$="CAL"THEN CO$=" Ca":V2=2
480 IF Q$="CAD"THEN CO$=" Cd":V2=2
490 IF Q$="BAR"THEN CO$=" Ba":V2=2
```

```
500 IF Q$="RAD"THEN CO$=" Ra":V2=2
510 IF MID$(N$,7,4)="ESTR"OR MID$(
    N$,11,4)="ESTR"THEN CO$=" Sr"
    :V2=2:GOTO 700
520 IF Q$="ZIN"OR Q$="CIN"THEN CO$
    =" Zn":V2=2
530 IF Q$="BER"THEN CO$=" Be":V2=2
540 IF Q$="MAG"THEN CO$=" Mg":V2=2
550 IF Q$="CUP"THEN CO$=" Cu":V1=1
    :V2=2
560 IF Q$="MER"THEN CO$=" Hg":V1=1
    :V2=2
570 IF Q$="ALU"THEN CO$=" Al":V2=3
580 IF Q$="BIS"THEN CO$=" Bi":V2=3
590 IF Q$="AUR"THEN CO$=" Au":V1=1
    :V2=3
600 IF Q$="FER"THEN CO$=" Fe":V1=2
    :V2=3
610 IF Q$="CRO"THEN CO$=" Cr":V1=2
    :V2=3
620 IF Q$="COB"THEN CO$=" Co":V1=2
    :V2=3
630 IF Q$="NIQ"THEN CO$=" Ni":V1=2
    :V2=3
640 IF Q$="MAN"THEN CO$=" Mn":V1=2
    :V2=3
650 IF Q$="PLA"THEN CO$=" Pt":V1=2
    :V2=4
660 IF Q$="EST"THEN CO$=" Sn":V1=2
    :V2=4
670 IF Q$="PLU"THEN CO$=" Pb":V1=2
    :V2=4
680 IF Q$="CAR"THEN CO$=" C":V2=2
690 IF Q$="NIT"THEN CO$=" N":V1=1:
    V2=2
700 IF D$="OSO"THEN B=V1
710 IF D$="ICO"THEN B=V2
720 IF SAL=1 THEN 1240
730 IF B=0 OR CO$=""THEN 310 ELSE
    AA=XMOD2:AB=BMOD2
740 IF AA=0 AND AB=0 THEN X=X/2:B=
    B/2
750 IF PE=1 THEN B=B+1
760 GOSUB 1550:LOCATE 12,13:PRINTC
    O$;X$;O$;B$:GOTO 320
770 '##TRATAMIENTO DE HIDROXIDOS##
780 O$=" (OH)":X=1:Q$=MID$(N$,11,3
    ):GOTO 390
790 '##TRATAMIENTO DE ANHIDRIDOS##
800 O$=" O":X=2:M$=MID$(N$,11,3):I
    $=M$
810 IF M$="HIP"THEN M$=MID$(N$,15,
    3)
820 IF M$="PER"THEN M$=MID$(N$,14,
    3)
830 GOSUB 1360
840 CO$=C$
850 IF I$="HIP"AND D$="OSO"THEN B=
    V1 ELSE IF D$="OSO"THEN B=V2
860 IF I$="PER"AND D$="ICO"THEN B=
    V4 ELSE IF D$="ICO"THEN B=V3
870 GOTO 730
880 '##TRATAMIENTO DE LOS ACIDOS##
890 M$=MID$(N$,7,3):I$=M$:DE$=RIGH
    T$(N$,5):H$="H":O$=" O":X=2:Y
    =2
900 IF M$="HIP"OR M$="MET"OR M$="P
    IR"OR M$="ORT"THEN M$=MID$(N$
    ,11,3)
910 IF M$="PER"THEN M$=MID$(N$,10,
    3)
920 GOSUB 1360
930 IF I$="HIP"AND D$="OSO"THEN V=
    V1 ELSE IF D$="OSO"THEN V=V2
940 IF I$="PER"AND D$="ICO"THEN V=
    V4 ELSE IF D$="ICO"THEN V=V3
950 IF DE$="DRICO"THEN 1290
960 IF V=0 THEN 310
970 ON KK GOTO 980,1030,1050,1090,
    1320
980 IF V=0 THEN 310
990 V=V+1
1000 AC=YMOD2:AD=XMOD2:AE=VMOD2:IF
     AC=0 AND AD=0 AND AE=0 THEN
     Y=Y/2:X=X/2:V=V/2
1010 IF SAL=1 THEN 1220
1020 GOSUB 1550:IF C$="" THEN 310
     ELSE LOCATE 12,13:PRINTH$;Y$;
     C$;X$;O$;V$:GOTO 320
1030 AF=VMOD2:AG=XMOD2:IF AF=0 AND
     AG=0 THEN V=V/2:X=X/2
1040 IF C$=" Mn" THEN 980 ELSE V=V
     +1:GOTO 1010
1050 IF I$<>"ORT"AND I$<>"MET"AND
     I$<>"PIR" THEN I$="ORT"
1060 IF I$="MET"THEN 980
1070 IF I$="PIR"THEN Y=4:V=V+2:GOT
     O 1000
1080 IF I$="ORT"THEN Y=6:V=V+3:GOT
     O 1000
```

```
1090 IF C$=" C"THEN 1030
1100 IF I$="ORT"OR I$="SIL"THEN 11
     10 ELSE IF I$="MET"THEN 1030
1110 IF D$="ICO"THEN Y=4:X=1:V=4::
     GOTO 1010 ELSE GOTO 310
1120 '##TRATAMIENTO DE LAS SALES##
1130 LOCATE 3,4:LINEINPUT "1a COMP
     ONENTE: ";N1$:LOCATE 3,7:LINE
     INPUT"2a COMPONENTE: ";N2$:D1
     $=RIGHT$(N1$,3):SAL=1:Y=2:O$=
     " O":X=2:D2$=RIGHT$(N2$,3):M$
     =LEFT$(N1$,3):I$=M$:M1$=LEFT$
     (N2$,3)
1140 IF N1$="HIDRURO"AND N2$="NITR
     ICO"THEN CO$="N":Y$=" ":C$="
     H":B$=" 3":GOTO 1240
1150 IF N1$="HIDRURO"AND N2$="CARB
     ONICO"THEN CO$="C":Y$=" ":C$=
     " H":B$=" 4":GOTO 1240
1160 IF M$="HIP"OR M$="MET"OR M$="
     PIR"OR M$="ORT"THEN M$=MID$(N
     1$,5,3) ELSE IF M$="PER"THEN
     M$=MID$(N1$,4,3)
1170 IF D1$="ATO"THEN D$="ICO"
1180 IF D1$="URO"THEN DE$="DRICO"
1190 IF D1$="ITO"THEN D$="OSO"
1200 GOSUB 1360
1210 GOTO 930
1220 N$="OXIDO "+N2$
1230 D$=D2$:O$=M1$:GOTO 390
1240 AI=YMOD2:AJ=BMOD2:IF AI=0 AND
      AJ=0 THEN Y=Y/2:B=B/2
1250 AI=YMOD3:AJ=BMOD3:IF AI=0 AND
      AJ=0 THEN Y=Y/3:B=B/3
1260 IF KE=1 THEN GOSUB 1550:IF CO
     $=""THEN 310 ELSE LOCATE 14,1
     3:PRINTCO$;Y$;C$;B$:GOTO 320
1270 GOSUB 1550:IF CO$=""THEN 310
1280 IF CO$="" OR C$="" THEN 310 E
     LSE LOCATE 10,13:PRINTCO$;Y$;
     "(";C$;X$;O$;V$;" )";B$:GOTO
     320
1290 IF KK=1 THEN 1300 ELSE IF KK=
     2 THEN 1310
1300 IF SAL=1 THEN KE=1:Y=1:GOTO 1
     220 ELSE LOCATE 14,13:PRINTH$
     ;" ";C$;" ":GOTO 320
1310 IF SAL=1 THEN KE=1:Y=2:GOTO 1
     220 ELSE LOCATE 14,13:PRINTH$
     ;2;C$;" ":GOTO 320
1320 IF I$="MET" AND D$="ICO" THEN
      Y=1:X=1:V=2:GOTO 1020
1330 IF(I$="ORT" OR I$="BOR")AND D
     $="ICO"THEN Y=3:X=1:V=3:GOTO
     1020
1340 GOTO 310
1350 REM ##VALENCIAS DE LOS NO-MET
     ALES##
1360 IF M$="FLU" THEN C$=" F":V1=1
     :V2=3:V3=5:V4=7:KK=1
1370 IF M$="HID" THEN C$=" H":KK=1
1380 IF M$="CLO"THEN C$=" Cl":V1=1
     :V2=3:V3=5:V4=7:KK=1
1390 IF M$="BRO"THEN C$=" Br":V1=1
     :V2=3:V3=5:V4=7:KK=1
1400 IF M$="IOD"THEN C$=" I":V1=1:
     V2=3:V3=5:V4=7:KK=1
1410 IF M$="SUL"THEN C$=" S":V1=2:
     V2=4:V3=6:V4=7:KK=2
1420 IF M$="SEL"THEN C$=" Se":V1=2
     :V2=4:V3=6:V4=7:KK=2
1430 IF M$="TEL"THEN C$=" Te":V1=2
     :V2=4:V3=6:V4=7:KK=2
1440 IF M$="NIT"THEN C$=" N":V2=3:
     V3=5:KK=1
1450 IF M$="FOS"THEN C$=$ P":V2=3:
     V3=5:KK=3
1460 IF M$="ARS"THEN C$=" As":V2=3
     :V3=5:KK=3
1470 IF M$="ANT"THEN C$=" Sb":V2=3
     :V3=5:KK=3
1480 IF M$="BOR"THEN C$=" B":V2=3:
     V3=5:KK=5
1490 IF M$="CAR"THEN C$=" C":V3=4:
     KK=4
1500 IF M$="SIL"THEN C$=" Si":V3=4
     :KK=4
1510 IF M$="MAN"THEN C$=" Mn":V2=4
     :V3=6:V4=7:KK=2
1520 IF M$="CRO" THEN C$=" Cr":V3=
     6:KK=2
1530 IF M$="BIS" THEN C$=" Bi":V3=
     5:KK=1
1540 RETURN
1550 X$=STR$(X):B$=STR$(B):Y$=STR$
     (Y):V$=STR$(V)
1560 IF X=1 THEN X$=" "
1570 IF B=1 THEN B$=" "
1580 IF Y=1 THEN Y$=" "
1590 IF V=1 THEN V$=" "
1600 RETURN
```

Registro QSL para radioaficionados

Todos los radioaficionados que dispongan de un ordenador, además del equipo en cuestión, pueden llevar su registro de QSL mediante este práctico y eficaz programa. El menú de opciones permite realizar las operaciones más diversas, desde la de registrar un QSO hasta la de obtener una salida en cassette de los datos. Las opciones son las siguientes;

1. Registrar un nuevo QSO.
2. Modificar registro.
3. Búsqueda de un indicativo.
4. Listado de datos.
5. Imprimir los 5 últimos QSO.
6. Salida de datos a cassette.

A la hora de obtener el listado de los QSO, lo podremos realizar por indicativo, modo, frecuencia, prefijo, fecha o simplemente, listarlo conjuntamente.



Aún tratándose de un buen programa, admite algunas mejoras, como por ejemplo, una opción de grabar los datos en disco para todos aquellos que dispongan de este periférico.

Manuel Angel Sirgo Ortea
Gijón

```
10 ' M.A.Sirgo Ortea para M S X
20 ' Libro de guardia
30 ' INICIALIZAR
40 CLEAR 4000
50 CLS:J=0:C1=31:KEY OFF
60 DIM KL$(250),DT$(250),UT$(25
   0),FQ$(250)
70 DIM MD$(250),PR$(250),RS$(25
   0),RR$(250)
80 DIM UE$(250),QS$(250),QR$(25
   0),CM$(250)
90 BL$=" !"
100 E1$=" MUY LARGO."
110 E2$=" CARACTERES MAX."
120 LOCATE 8,5:PRINT"* LIBRO DE
    GUARDIA *"
130 PRINT TAB(8)"* LIBRO DE GUA
    RDIA *"
140 PRINT TAB(8)"* LIBRO DE GUA
    RDIA *"
150 LOCATE 2,20:PRINT"DESEA INI
    CIALIZAR UN FICHERO (S/N)?
    "
160 A$=INPUT$(1):IF A$="S" THEN
     180
170 IF A$="N" THEN 460 ELSE 160
180 ' INICIALIZAR FICHERO
190 CLS:PRINT:PRINTTAB(5)"INICI
    ALIZAR FICHERO":PRINT
200 GOSUB 900:IN$=KX$
210 GOSUB 940
220 INPUT"NUMERO DE REGISTRO";N
    R:ZZ=1
230 ' CARATULA
240 CLS:LOCATE 8,5:PRINT"LIBRO
    DE GUARDIA DE:";IN$
```

```
250 TT=LEN(IN$):LOCATE28,6
260 FOR F=1 TO TT:PRINT"‾";:NEXT F
270 PRINT:PRINT:PRINT TAB(2) "REGISTRO NUMERO ";NR:PRINT
280 IF ZZ<>1 THEN 290 ELSE PRINTTAB(2)"CREADO EL :";DA$:GOTO 320
290 PRINTTAB(2)"CREADO EL:";DT$(1)
300 PRINT:PRINT TAB(2) "ENTRADAS -";J
310 PRINT:PRINT TAB(2) "ULTIMA ENTRADA:";DT$(J)
320 LOCATE 5,20:PRINT"PULSE UNA TECLA PARA CONTINUAR"
330 A$=INPUT$(1):IF A$<>"" THEN 340 ELSE 330
340 ' MENU PRINCIPAL
350 CLS:LOCATE 5,2:PRINT"PULSE NUMERO DE OPCION:"
360 PRINT:PRINT:PRINT TAB(5) "1.- REGISTRAR NUEVO QSO.":PRINT
370 PRINT TAB(5) "2.- MODIFICAR REGISTRO.":PRINT
380 PRINT TAB(5) "3.- BUSQUEDA DE UN INDICATIVO.":PRINT
390 PRINT TAB(5) "4.- LISTADO DE DATOS.":PRINT
400 PRINT TAB(5) "5.- IMPRIMIR 5 ULTIMOS QSO.":PRINT
410 PRINT TAB(5) "6.- SALIDA DE DATOS A CASETTE."
420 A$=INPUT$(1):AO=VAL(A$)
430 IF AO<1 OR AO>6 THEN 420
440 ON AO GOSUB 630,1220,2650,1690,2430,2510
450 GOTO 340
460 'ENTRADA DATOS CASETTE
470 CLS:M=0
480 PRINT"   COLOQUE LA CINTA DE DATOS Y PULSE":PRINT
490 PRINT"<< ENTER >> CUANDO TENGA EL CASETTE":PRINT
500 INPUT"LISTO";NO$
510 PRINT:PRINT"LEYENDO...":MOTOR ON
520 OPEN"CAS:QSL" FOR INPUT AS1
530 INPUT#1,J:INPUT#1,IN$:INPUT#1,NR
540 IF EOF(1) THEN 610
550 M=M+1
560 INPUT#1,KL$(M):INPUT#1,DT$(M):INPUT#1,UT$(M)
570 INPUT#1,FQ$(M):INPUT#1,MD$(M):INPUT#1,PR$(M)
580 INPUT#1,RS$(M):INPUT#1,RR$(M):INPUT#1,UE$(M)
590 INPUT#1,QS$(M):INPUT#1,QR$(M):INPUT#1,CM$(M)
600 GOTO 540
610 CLOSE:MOTOR OFF
620 GOTO 230
630 ' REGISTRO NUEVO QSO
640 J=J+1:CLS
650 GOSUB 900:KL$(J)=KX$
660 GOSUB 940:DT$(J)=DA$
670 PRINT"Cual es la hora de comienzo";:GOSUB 1060:UT$(J)=U$
680 INPUT"Cual es la frecuencia?";FQ$(J)
690 IF LEN(FQ$(J))>8 THEN PRINT BL$;E1$;8;E2$:GOTO 680
700 INPUT"Cual es el modo?";MD$(J)
710 IF LEN(MD$(J))>3 THEN PRINT BL$;E1$;3;E2$:GOTO 700
720 INPUT"Cual es la potencia?";PR$(J)
730 IF LEN(PR$(J))>4 THEN PRINT BL$;E1$;4;E2$:GOTO 720
740 INPUT"Cual es su RST?";RS$(J)
750 IF LEN(RS$(J))>3 THEN PRINT BL$;E1$;3;E2$:GOTO 740
760 INPUT"Cual es mi RST?";RR$(J)
770 IF LEN(RR$(J))>3 THEN PRINT BL$;E1$;3;E2$:GOTO 760
780 PRINT"Cual es la hora de final QSO?";:GOSUB 1060:UE$(J)=U$
790 INPUT"Enviare QSL (E)?";QS$(J)
800 IF LEN(QS$(J))>1 THEN PRINT BL$;E1$;1;E2$:GOTO 790
810 PRINT"Entra un comentario(";C1;"caracteres max)";
```

```
820 INPUT CM$(J)
830 IF LEN(CM$(J))>C1 THEN PRIN
    TBL$;LEN(CM$)-C1;"Demasiad
    os caracteres.Repite":GOTO
     820
840 LOCATE 1,20:PRINT"SEÑALE OP
    CION:"
850 PRINT"1.- NUEVO QSO  -  2.-
     MENU"
860 A$=INPUT$(1)
870 IF A$="1" THEN 630
880 IF A$="2" THEN RETURN
890 GOTO 860
900 ' ENTRADA INDICATIVO
910 CLS:INPUT"Indicativo del co
    rresponsal?";KX$
920 IF LEN(KX$)>10 THEN PRINTBL
    $;E1$;10;E2$:GOTO 900
930 RETURN
940 ' ENTRADA FECHA
950 INPUT"Cual es la fecha(DD/M
    M/AA)?";DA$
960 IF LEN(DA$)<6 THEN 1050
970 IF MID$(DA$,3,1)="/" THEN 1
    000
980 IF MID$(DA$,2,1)<>"/" THEN
    1050
990 DA$="0"+DA$
1000 IF MID$(DA$,6,1)="/" THEN
     1030
1010 IF MID$(DA$,5,1)<>"/" THEN
     1050
1020 DA$=LEFT$(DA$,3)+"0"+RIGHT
     $(DA$,4)
1030 IF LEN(DA$)<>8 THEN 1050
1040 RETURN
1050 PRINTBL$;"ERROR.REPITE.":G
     OTO 940
1060 ' ENTRADA HORA
1070 INPUT"(HHMM)?";U$
1080 IF LEN(U$)>4 THEN PRINTBL$
     ;E1$;4;E2$;"Repite":GOTO 1
     060
1090 IF U$="" THEN RETURN
1100 IF LEN(U$)<4THEN FORH=LEN(
     U$)+1 TO 4:U$="0"+U$:NEXT
     H
1110 RETURN
1120 ' PRESENTACION CABECERA
1130 CLS:PRINT" Nº  INDICATIVO
      HORA  POT.  RST  QSL"
1140 PRINT" QSL    FECHA    FRE
     C. MODO (S/M) E/R"
1150 PRINT" --- ----------  ---
     -- ---- ----- --- "
1160 RETURN
1170 ' ESCRIBIR QSO
1180 PRINT F;TAB(6) KL$(F);TAB(
     17) UT$(F);TAB(24)PR$(F);T
     AB(30) RS$(F);TAB(35) QS$(
     F)
1190 PRINTTAB(6)DT$(F);TAB(17)F
     Q$(F);TAB(24)MD$(F);TAB(30
     )RR$(F);TAB(35)QR$(F)
1200 PRINTTAB(6)"*.";CM$(F)
1210 PRINT:RETURN
1220 'EDITAR UN REGISTRO A MODI
     FICAR
1230 CLS:A$=""
1240 PRINT"ENTRA EL NUMERO DE R
     EGISTRO A EDITAR O"
1250 INPUT"DA << ENTER >> PARA
     EDITAR EL ULTIMO.";A$
1260 IF A$="" THEN F=J:GOTO 129
     0
1270 F=VAL(A$)
1280 IF F>J OR F<1 THEN 1220
1290 GOSUB 1120:GOSUB 1170
1300 PRINT"DESEA CORREGIRLO (S/
     N)?"
1310 A$=INPUT$(1)
1320 IF A$="S" THEN 1350
1330 IF A$="N" THEN 340
1340 GOTO 1310
1350 PRINT"DA << ENTER >> PARA
     CONSERVAR LA INFOR-MACION"
     ;
1360 PRINT" O UN NUEVO VALOR PA
     RA CAMBIARLO."
1370 PRINT"FECHA:";DT$(F);:INPU
     T DA$:IF DA$<>"" THEN GOSU
     B 960:DT$(F)=DA$
1380 PRINT"HORA DE COMIENZO:";U
     T$(F);:INPUT U$:IF U$<>""
     THEN GOSUB 1080:UT$(F)=U$
1390 PRINT"FRECUENCIA:";FQ$(F);
     :INPUT A$:IF A$="" THEN 14
     20
1400 IF LEN(A$)>8 THEN PRINT BL
     $;E1$;8;E2$:GOTO 1390
```

```
1410 FQ$(F)=A$
1420 PRINT"MODO:";MD$(F);:INPUT A$:IF A$="" THEN 1450
1430 IF LEN(A$)>3 THEN PRINT BL$;E1$;3;E2$:GOTO 1420
1440 MD$(F)=A$
1450 PRINT"POTENCIA:";PR$(F);:INPUT A$:IF A$="" THEN 1480
1460 IF LEN(A$)>4 THEN PRINT BL$;E1$;4;E2$:GOTO 1450
1470 PR$(F)=A$
1480 PRINT"SU RST:";RS$(F);:INPUT A$:IF A$="" THEN 1510
1490 IF LEN(A$)>3 THEN PRINT E1$;3;E2$:GOTO 1480
1500 RS$(F)=A$
1510 PRINT"MI RST:";RR$(F);:INPUT A$:IF A$="" THEN 1540
1520 IF LEN(A$)>3 THEN PRINT E1$;3;E2$:GOTO 1510
1530 RR$(F)=A$
1540 PRINT"HORA FINAL DEL QSO (QRT):";UE$(F);:INPUT U$:IF U$<>"" THEN GOSUB 1080:UE$(F)=U$
1550 PRINT"ENVIADA QSL (E):";QS$(F);:INPUT A$:IF A$="" THEN 1580
1560 IF LEN (A$)<>1 THEN PRINT BL$;E1$;1;E2$:GOTO 1550
1570 QS$(F)=A$
1580 PRINT"RECIBIDA QSL (R):";QR$(F);:INPUT A$:IF A$="" THEN 1610
1590 IF LEN (A$)<>1 THEN PRINT BL$;E1$;1;E2$:GOTO 1580
1600 QR$(F)=A$
1610 PRINT"COM.";CM$(F);:INPUT"...?";A$:IF A$="" THEN 1640
1620 IF LEN(CM$)>C1 THEN PRINT BL$;E1$;C1;E2$:GOTO 1610
1630 CM$(F)=A$
1640 PRINT"CORRECTO (S/N)?"
1650 A$=INKEY$
1660 IF A$="N" THEN 1290
1670 IF A$="S" THEN RETURN
1680 GOTO 1650
1690 ' MENU LISTADO DE DATOS
1700 CLS:LOCATE 3,3:PRINT"LISTADO DE DATOS POR ..."
1710 PRINT:PRINTTAB(5)"1.- INDICATIVO."
1720 PRINT:PRINTTAB(5)"2.- MODO."
1730 PRINT:PRINTTAB(5)"3.- FRECUENCIA."
1740 PRINT:PRINTTAB(5)"4.- PREFIJO."
1750 PRINT:PRINTTAB(5)"5.- FECHA."
1760 PRINT:PRINTTAB(5)"6.- QSL'S NO RECIBIDAS."
1770 PRINT:PRINTTAB(5)"7.- TODOS LOS QSO."
1780 LOCATE 5,20:PRINT"PULSE NUMERO DE OPCION"
1790 A$=INPUT$(1):AO=VAL(A$)
1800 IF AO<1 OR AO>7 THEN 1790
1810 ON AO GOTO 1820,1880,2120,2260,2020,1960,2220
1820 'LISTADO POR INDICATIVO
1830 CLS:GOSUB 900:GOSUB 1120
1840 FOR F=1 TO J
1850 IF KL$(F)=KX$ THEN GOSUB 1170
1860 NEXT F
1870 GOTO 2350
1880 'LISTADO POR MODO
1890 CLS:INPUT"CUAL ES EL MODO";M$
1900 IF LEN(M$)>3 THEN PRINTBL$;E1$;3;E2$:GOTO 1890
1910 GOSUB 1120
1920 FOR F=1 TO J
1930 IF MD$(F)=M$ THEN GOSUB 1170
1940 NEXT F
1950 GOTO 2350
1960 'LISTADO POR QSL'S NO RECIBIDAS
1970 GOSUB 1120
1980 FOR F=1 TO J
1990 IF QR$(F)<>"R" THEN GOSUB 1170
2000 NEXT F
2010 GOTO 2350
2020 'LISTADO POR FECHAS
2030 CLS:INPUT"FECHA INICIAL(DD/MM/AA)";DA$:GOSUB 960:SD$=MID$(DA$,7,2)+MID$(DA$,4,
```

```
      2)+MID$(DA$,1,2)
2040 INPUT"FECHA FINAL(DD/MM/AA
      )";DA$:GOSUB 960:ED$=MID$(
      DA$,7,2)+MID$(DA$,4,2)+MID
      $(DA$,1,2)
2050 IF SD$>ED$ THEN PRINTBL$;"
      ERROR":GOTO 2030
2060 GOSUB 1120
2070 FOR F=1 TO J
2080 PT$=MID$(DT$(F),7,2)+MID$(
      DT$(F),4,2)+MID$(DT$(F),1,
      2)
2090 IF PT$>=SD$ AND PT$<=ED$ T
      HEN GOSUB 1170
2100 NEXT F
2110 GOTO  2350
2120 'LISTADO POR FRECUENCIAS
2130 CLS:INPUT"CUAL ES LA FREC.
       MENOR";LF
2140 CLS:INPUT"CUAL ES LA FREC.
       MAYOR";UF
2150 IF UF<LF THEN PRINTBL$;"ER
      ROR":GOTO 2130
2160 GOSUB 1120
2170 FOR F=1 TO J
2180 FE=VAL(FQ$(F))
2190 IF FE>=LF THEN IF FE<=UF T
      HEN GOSUB 1170
2200 NEXT F
2210 GOTO  2350
2220 'LISTADO DE TODOS LOS QSO
2230 GOSUB 1120
2240 FOR F=1 TO J
2250 GOSUB 1170
2260 NEXT F
2270 GOTO  2350
2280 'LISTADO POR PREFIJO
2290 CLS:PRINT"CUAL ES EL PREFI
      JO (2 LETRAS)"
2300 T$=INPUT$(2)
2310 GOSUB 1120
2320 FOR F=1 TO J
2330 IF MID$(KL$(F),1,2)=T$ THE
      N GOSUB 1170
2340 NEXT F
2350 'OPCION LISTADO
2360 PRINT:PRINT:PRINT
2370 LOCATE 1,20:PRINT"SEÑALE O
      PCION:"
2380 PRINT"1.- LISTADO DE DATOS
        -  2.- MENU"
2390 A$=INPUT$(1)
2400 IF A$="1" THEN 1690
2410 IF A$="2" THEN RETURN
2420 GOTO 2390
2430 ' 5 ULTIMOS QSO
2440 GOSUB 1120:
2450 FOR F=J-4 TO J
2460 GOSUB 1170
2470 NEXT F
2480 PRINT:PRINT:PRINT
2490 LOCATE 5,20:PRINT"PULSE UN
      A TECLA PARA CONTINUAR"
2500 A$=INPUT$(1):IF A$<>"" THE
      N 340 ELSE 2500
2510 'GRABANDO
2520 CLS:PRINT TAB(8)"- GRABAR
      EN CASETTE -":PRINT:PRINT
2530 PRINT"PULSE <<ENTER>> CUAN
      DO TENGA EL CASETTE"
2540 INPUT"LISTO";NO$
2550 PRINT:PRINT"GRABANDO...":M
      OTOR ON
2560 OPEN"CAS:QSL" FOR OUTPUT A
      S1
2570 PRINT#1,J:PRINT#1,IN$:PRIN
      T#1,NR
2580 FOR M=1 TO J
2590 PRINT#1,KL$(M):PRINT#1,DT$
      (M):PRINT#1,UT$(M)
2600 PRINT#1,FQ$(M):PRINT#1,MD$
      (M):PRINT#1,PR$(M)
2610 PRINT#1,RS$(M):PRINT#1,RR$
      (M):PRINT#1,UE$(M)
2620 PRINT#1,QS$(M):PRINT#1,QR$
      (M):PRINT#1,CM$(M)
2630 NEXT M
2640 CLOSE:MOTOR OFF:RETURN
2650 'BUSQUEDA UN INDICATIVO
2660 GOSUB 900:GOSUB 1120
2670 FOR F=1 TO J
2680 IF KX$=KL$(F) THEN GOSUB 1
      170:GOTO 2700
2690 NEXT F
2700 LOCATE 5,20:PRINT"SEÑALE O
      PCION:"
2710 PRINT"1. BUSCAR OTRO INDIC
      ATIVO. - 2. MENU"
2720 A$=INPUT$(1)
2730 IF A$="1" THEN 2650
2740 IF A$="2" THEN RETURN
2750 GOTO 2720
```



Leer un carácter en la pantalla

Para leer un carácter de una determinada posición de la pantalla, bastará con utilizar la instrucción DEF FN definida de la manera siguiente.

10 DEF FN SC (X,Y) = VPEEK (X+2+Y*40)

El programa siguiente ilustra el uso de esta función. PROG. 1.

```
PROGRAMA 1: COPYCHR$

10 DEF FN SC(X,Y)=VPEEK(X+2+Y*40)
20 SCREEN 0:WIDTH 37:KEY OFF
30 LOCATE 3,10:PRINT"PRUEBA DE COPIA DE CARACTERES"
40 LOCATE 9,21:PRINT"PULSA UNA TECLA"
50 IF INKEY$="" THEN 50
55 LOCATE 9,21:PRINTSPC(20)
60 FOR X=1 TO 37
70 A=FN SC(X,10)
80 LOCATE X,0:PRINTCHR$(A)
90 FOR P=1 TO 90:NEXT P
100 NEXT X
```

Sonidos con MSX

Para todos los aficionados a los sonidos extraños, presentamos un corto programa con el que podrán modificar a su gusto todos los valores de los registros del chip de sonido. PROG. 2.

```
PROGRAMA 2: SONIDOS

10 SCREEN 0:WIDTH 37:KEY OFF
20 FOR A=0 TO 13
30 LOCATE 0,A:PRINT "valor del
registro";A;:INPUT X
40 IF X>255 THEN 30
45 SOUND A,X
50 NEXT A
60 GOTO 20
```





BASIC PARA PRINCIPIANTES

Siendo uno de los lenguajes de programación más potente en el mercado de los ordenadores personales, el BASIC del MSX esconde misterios que el profano de la materia habrá podido observar.

El editor de pantalla

Antes de escribir un programa, veremos algunas utilidades del EDITOR DE PANTALLA del ordenador MSX.

El editor de pantalla es un conjunto de códigos que nos permite escribir un programa y modificarlo. Los ordenadores MSX llevan incorporadas las siguientes teclas cuyas funciones se indican a continuación:

— *TAB:* Mueve el cursor 8 posiciones a la derecha y borra todo lo que se encuentra en su camino.

— *HOME:* Sitúa el cursor en la posición superior izquierda de la pantalla.

— *STOP:* Detiene temporalmente el listado o la ejecución de un programa.

— *INS:* Sirve para insertar carácteres a la izquierda de la posición del cursor.

— *DEL:* Borra el carácter donde está situado el cursor.

— *BS:* Mueve el cursor una posición a la izquierda borrando el carácter correspondiente.

— *↑:* Mueve el cursor hacia arriba.

— *↓:* Mueve el cursor hacia abajo.

— *←:* Mueve el cursor hacia la izquierda.

— *→:* Mueve el cursor hacia la derecha.

— *RETURN:* Validación de línea: se pulsará al final de cada línea para que el ordenador la almacene como válida.

Además, pulsando simultáneamente las teclas *SHIFT* y *HOME,* se borra la pantalla y el cursor se sitúa en la posición superior izquierda.

Las teclas *ESC* y *SELECT* no tienen ninguna acción.

Por otra parte, al pulsar simultáneamente las teclas *CTRL* (Control) y una de las teclas alfabéticas, se consiguen los siguientes efectos: (El número de la izquierda corresponde al código *ASCII* del par de teclas pulsado).

2 *CTRL+B:* Traslada el cursor al

principio de la palabra anterior.

3 *CTRL+C:* Interrupción: Interrumpe totalmente el listado o la ejecución de un programa.

5 *CTRL+E:* Borra todo lo que hay a la derecha del cursor en la línea actual.

6 *CTRL+F:*Traslada el cursor al principio de la palaba siguiente.

7 *CTRL+G:* Emite un pitido (*BEEP*).

8 *CTRL+H:* Equivale a *BS.*

9 *CTRL+I:* Equivale a *TAB.*

10 *CTRL+J:* Mueve el cursor una línea hacia abajo.



11 *CTRL+K:* Equivale a *HOME.*

12 *CTRL+L:* Equivale a *SHIFT+HOME.*

13 *CTRL+M:* Equivale a *RETURN.*

14 *CTRL+N:* Traslada el cursor al final de la línea actual.

18 *CTRL+R:* Equivale a *INS.*

21 *CTRL+U:* Borra la línea donde se encuentra el cursor.

24 *CTRL+X:* Equivale a *SELECT.*

27 *CTRL+[:* Equivale a *ESC.*

28 *CTRL+/:* Equivale a →.

29 *CTRL+]:* Equivale a ←.

30 *CTRL+ˆ* o *CTRL+~:* Equivale a ↑.

31 *CTRL+ –:* Equivale a ↓.

CTRL+STOP: Equivale a *CTRL+C.*

Empecemos con un programa

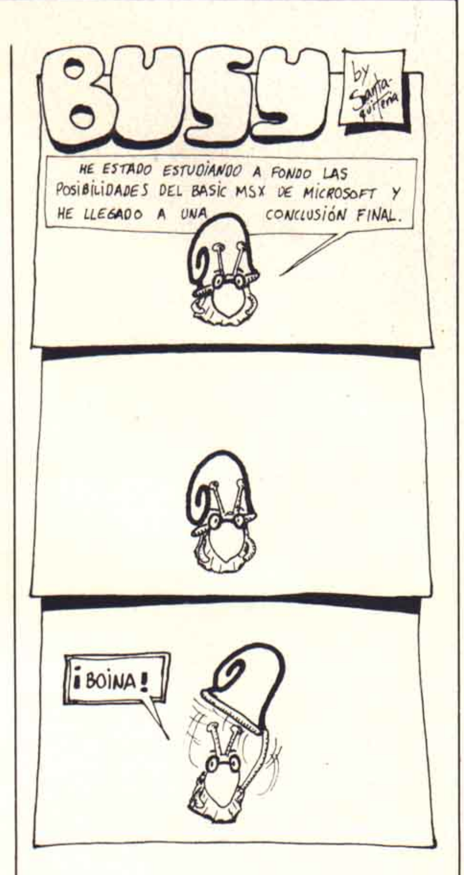
Lo que tu pretendes desde el momento de comprar tu ordenador MSX es poder realizar tus propios programas, es decir, poder preparar el ordenador para que haga repetidas veces lo que desees. Pues bien, para ir conociendo cómo programar en *BASIC* iremos directamente al grano... y empezaremos programando.

Observa la figura 4. Efectivamente se trata de un programa,... que vamos a ir desmenuzando:

La comunicación con el ordenador debe ser de una forma estructurada. No basta con que le digas qué debe hacer, sino que le tienes que decir en qué orden. Por eso se utilizan los números de línea, de forma que el programa quede completamente estructurado. El programa correrá exactamente en el orden en que estén numeradas las líneas (excepto en las bifurcaciones).

En las líneas 10, 20 y 30 verás la palabra *REM.* Aparentemente no significa nada, pero en realidad se trata de una INSTRUCCION o, lo que es lo mismo, una palabra con la que decimos a *BASIC* que haga algo. *BASIC* tiene muchas palabras que, como *REM,* tienen un significado especial. Iremos viendo cuáles son y para qué sirven estas PALABRAS RESERVADAS.

Con la instrucción *REM* (en inglés REMARK) el ordenador entiende que todo lo que viene a continuación es una ANOTACION o COMENTARIO del usuario, que no significa nada para BASIC, y por tanto debe ignorar y saltar a la línea siguiente. Esto nos permite escribir aclaraciones (líneas 90 y 100) o, por ejemplo, el título del programa (líneas 10, 20 y 30).



En la línea 40 hemos escrito la instrucción *SCREEN* 0,,0. *SCREEN* significa PANTALLA. El ordenador MSX puede presentar la pantalla de 4 formas distintas. *SCREEN 0* es una de esas formas; es la presentación de pantalla en MODO TEXTO DE 40 COLUMNAS. Con este modo de pantalla se pueden escribir textos (no se puede dibujar) con una anchura máxima de 40 columnas, es decir, de 40 caracteres por línea. Si en lugar de *SCREEN 0* pones *SCREEN 1,* el ordenador presentará la pantalla en

MODO TEXTO DE 32 COLUMNAS. Este modo tiene una serie de ventajas respecto al modo 0, de las que hablaremos más adelante. También conoceremos los MODOS GRAFICOS 2 y 3.

Después de *SCREEN 0* hemos escrito dos comas y otro cero. Lo que puede ponerse entre esas dos comas no nos interesa de momento. El úitimo cero elimina el chasquido de teclas. Así, después de *SCREEN 0,,0* al pulsar cualquier tecla no se oirá ningún chasquido. Si escribes *SCREEN,, 1* éste se restablece.

SCREEN además, actualiza la pantalla, borrando todo lo que haya escrito en ella.

La instrucción *WIDTH* de la línea 50 nos permite elegir, dentro del modo de texto, el ANCHO de pantalla que queramos. En el modo texto de 40 columnas *(SCREEN 0)* puedes escribir *WIDTH 40* ó cualquier número menor que 40. Te aconsejamos utilizar como máximo *WIDTH 39,* ya que con *WIDTH 40* no podrás ver algunos caracteres en las esquinas superiores e inferiores de la pantalla.

En el modo de texto de 32 columnas, el máximo valor que puedes dar a *WIDTH* es 32 *(WIDTH 32).* Si das un valor mayor, tendrás un error del tipo *ILLEGAL FUNCTION CALL.* Lo mismo ocurre si das un valor mayor de 40 en el modo 0. *WIDTH,* además, actualiza la pantalla.

CLS (en inglés CLEAR SCREEN) es otra instrucción y sirve para BORRAR la pantalla y situar el cursor en la esquina superior izquierda. En realidad esta instrucción sobra en nuestro programa, ya que *SCREEN* tiene el mismo efecto.

Como has visto en el encabezamiento, (líneas 10 y 20) nuestro programa sirve para calcular la longitud de una circunferencia y el área de un círculo. Ya sabes que la longitud de una circunferencia viene dada por:

$$LGT = 2\pi R$$

donde *LGT* es la longitud y *R* el radio. El área del círculo viene dada por:

$$AREA = \pi R^2$$

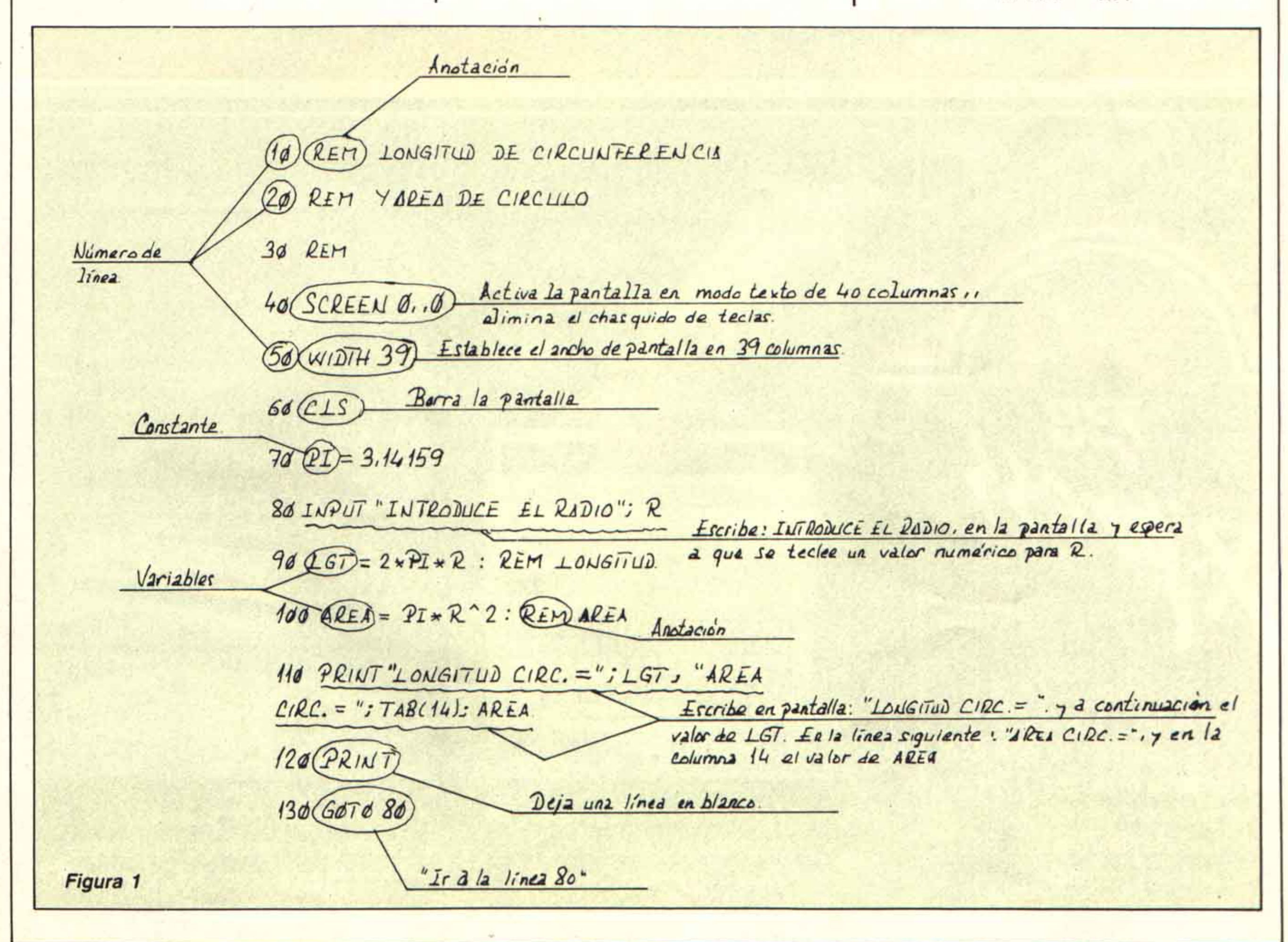


Figura 1

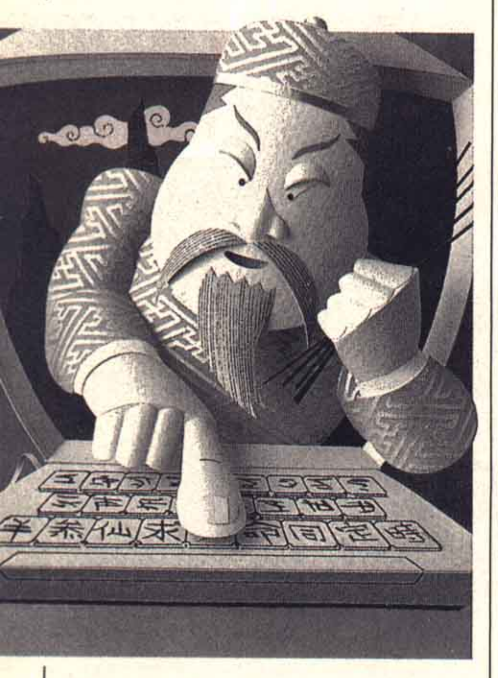
siendo AREA el área del círculo.

En ambos casos tenemos la constante π (pi) que es igual a 3.14159.

Nuestro programa tendrá por tanto que:

1. Pedirnos el valor del radio (R).
2. Hallar el valor de la longitud (LGT).
3. Hallar el valor del área (AREA).
4. Escribir los valores de la longitud y del área.
5. Empezar otra vez en el punto 1.

Fíjate en la línea 70. El ordenador no tiene dentro el número π



(pi), y por tanto nosotros lo hemos definido aquí. Le hemos dicho que dé a la variable PI el valor 3.14159, con lo que hemos convertido a PI en una CONSTANTE para nuestro programa.

Observa que los dos puntos de las líneas 90 y 100 sirven para separar las fórmulas de la izquierda de las respectivas aclaraciones *(REM)* de la derecha.

La instrucción *INPUT* de la línea 80 hace que el ordenador se pare en espera de que le des un valor (o varios). Fíjate en la estructura de esta instrucción: después de *INPUT* puedes poner lo que quieras entre comillas. Esto se escribirá tal cual en la pantalla. A continuación debes poner un punto y coma al que seguirá el nombre de la variable o variables que debes introducir desde el teclado.

Todo lo que escribas entre comillas después de *INPUT* y antes del punto y coma, se escribirá en la pantalla. Esto te ayudará a saber qué variable te está pidiendo el ordenador. Puedes escribir simplemente:

INPUT R

con lo que sólo aparecerá un signo de interrogación.

Imagínate que vas a hacer un programa que calcule el volumen de un cilindro:

$$V = \pi R^2 H$$

donde V es el volumen y H la altura.

Para introducir los datos (R y H) puedes escribir:

```
80 INPUT R
82 INPUT H
```

o bien:

```
80 INPUT "RADIO";R
82 INPUT "ALTURA";H
```

o bien:

```
80 INPUT R,H
```

o bien:

```
80 INPUT "RADIO,ALTURA";R,H
```

Todas estas formas son igualmente válidas. Las más claras son la segunda y la última, porque escriben en la pantalla qué variables debes introducir y en qué orden.

La última forma es la más condensada. Consta de un texto entre comillas que será escrito en la pantalla, un punto y coma y una lista de variables separadas por comas. El punto y coma y las comas se llaman SEPARADORES y sirven para separar el texto de las variables y para separar las variables respectivamente.

En general, la instrucción *INPUT* puede ser:

INPUT "Texto";A,B,C,D...

pudiendo admitir tantas variables como te permita el número máximo de carácteres que admite una línea (255).

En las líneas 90 y 100 calculamos la longitud de la circunferencia y el área del círculo. Fíjate que el signo de multiplicar se escribe en *BASIC* con un asterisco (*) y el signo de potencia con una tilde (^). En los ordenadores MSX, el signo * se consigue pulsando simultáneamente las teclas *SHIFT* y 8, y el ^ pulsando SHIFT+6.

LGT, AREA y R son VARIABLES, porque varían a lo largo del programa. Una variable puede ser una palabra tan larga como quieras (por ejemplo AREA tiene 4 carácteres), pero *BASIC* sólo toma los 2 primeros caracteres; el resto pueden ayudarte a reconocer la variable. (Por ejemplo, AREA y ARCO son lo mismo —AR— para el *BASIC* MSX). Toda variable debe empezar por una letra y luego puede tener una letra o un número. Puedes escribirlas en minúsculas, pero el ordenador las escribirá automáticamente en mayúsculas. No puedes utilizar dentro de una variable una palabra reservada. Por ejemplo, no puedes escribir *LONG* en lugar de *LGT* ya que *ON* es una palabra reservada y te daría error.

La instrucción *PRINT* significa ESCRIBIR EN LA PANTALLA. Con esta instrucción, todo lo que pongas entre comillas se escribe tal

cual. Este es el caso de "LONGITUD CIRC.=" o "AREA CIRC.=", en nuestro programa. Por supuesto *PRINT* también escribe el valor de una variable, por ejemplo *LGT* y AREA en nuestro programa. *PRINT,* como *INPUT,* también admite separadores. Estos pueden ser el punto y coma o la coma.

El punto y coma hace que lo que viene a continuación se imprima inmediatamente a partir de la posición del cursor.

La coma hace que lo que viene a continuación se imprima a partir de la columna 14 de la línea donde está el cursor, si éste se encuentra a la izquierda de dicha columna, o al principio de la línea siguiente, si el cursor está a la derecha de la misma.



Dentro de la instrucción *PRINT,* se pueden utilizar tabuladores mediante la función *TAB*(x). Esta función hará que el cursor se sitúe en la columna x de la fila acutal (la columna de la izquierda corresponde a x=0) si estaba a la izquierda de dicha posición. En caso contrario, el cursor no se moverá.

En general, la instrucción *PRINT* será:

PRINT "Texto";A,"Texto";B,C...

donde lógicamente, "Texto" puede ser cualquier texto encerrado entre comillas, las variables A,B,C... pueden estar situadas donde se desee. Si al final de la instrucción *PRINT* no se escribe ningún separador, el cursor se situará al principio de la línea siguiente (avance de carro).

En resumen, lo que hará la instrucción *PRINT* de la línea 110 será lo siguiente: primero escribirá la serie de carácteres *"LONGITUD CIRC.=";* justo a continuación, escribirá el valor de *LGT.* En la línea siguiente escribirá: ·AREA CIRC.= ", y a continuación el valor de AREA justo debajo de *LGT.* Si el valor del radio (R) es 5, el de la longitud será 31.4159 y el del área 78.53975, PRINT escribirá:

```
LONGITUD CIRC.= 31.4159
AREA CIRC.=     78.53975
```

Al final de la línea 110 no hemos puesto ningún separador, y por tanto el cursor se sitúa al principio de la línea siguiente.

La instrucción *PRINT* de la línea 120 hace que el cursor descienda una línea más y, por tanto, sirve para dejar una línea en blanco antes de volver a escribir nada.

Por último, la instrucción *GOTO 80* de la línea 130 es una sentencia de control que llamaremos BIFURCACION INCONDICIONAL, porque cada vez que el programa pase por ella, se bifurcará automáticamente a la línea 80. Con esto, el ordenador se prepara para recibir un nuevo valor de R.

Para interrumpir el programa deberás pulsar *CTRL+STOP.*

Resumamos esquemáticamente lo que hemos visto:

— *REM:* Anotaciones. El programa la ignora.

— *SCREEN 0,,0:* Prepara la pantalla en el modo texto de 40 columnas. Elimina el chasquido de teclas. Borra la pantalla.

— *WIDTH 39:* Establece una anchura de 39 columnas por línea.

— *CLS:* Borra la pantalla.

— *INPUT* "Texto"A,B...: Muestra "Texto" en la pantalla y espera a recibir valores para las variables A,B...

— *PRINT* "Texto";A,"Texto",B...: Imprime en la pantalla los textos sueltos entre comillas y los valores de las variables.

— *GOTO 80:* Bucle incondicional: Ir a la línea 80.

NOTA: Puedes grabar tus programas en cinta de cassette con la instrucción *CSAVE"Nombre",* en la que "Nombre" debe tener un máximo de 6 carácteres y empezar por una letra.

Para recuperar el programa de la cinta, hazlo mediante la instrucción *CLOAD"Nombre"* o, simplemente *CLOAD* (con ésta, el ordenador tomará el primer archivo que encuentre).



Sistemas de ecuaciones lineales

En el último número de MSX Magazine hablábamos del interés que representa, dentro de la ciencia y de la técnica, el disponer de un método potente para resolver sistemas de ecuaciones lineales. Estos sistemas se presentan, con cierta frecuencia, particularmente en ciertas ramas de la técnica.

Hablábamos entonces del método no iterativo de *Gauss-Jordan.* En esta ocasión hablaremos de algunos métodos iterativos. Estos presentan ciertas ventajas (práctica ausencia de errores de redondeo), y están indicados especialmente cuando se cumplen ciertas condiciones particulares. En cualquier caso, suele ser útil el uso de, al menos, dos métodos distintos, como medio de comprobación de la fiabilidad de ellos.

El método de Jacobi

El método de *Jacobi* es un método iterativo utilizable cuando el sistema de ecuaciones es tal que, en la matriz de los coeficientes, los elementos de la diagonal principal son dominantes, en valor absoluto, frente al resto de los elementos. Cuanto más dominantes sean los citados elementos de la diagonal principal, tanto más rápida será la convergencia del método.

Supongamos el siguiente sistema de 4 ecuaciones con 4 incógnitas:

(1)

$$\begin{aligned} 10x_1 + x_2 + x_3 + x_4 &= 19 \\ 2x_1 + 5x_2 + x_3 - x_4 &= 11 \\ -x_1 - x_2 + 4x_3 + x_4 &= 13 \\ x_1 - x_2 - 2x_3 + 6x_4 &= 17 \end{aligned}$$

Efectivamente, el sistema se puede resolver por el método de *Gauss-Jordan.* Sin embargo, en muchos casos el método de *Jacobi* puede ser más rápido que aquél. No es este el caso, ya que el cálculo del sistema (1) con ordenador por el método de *Gauss-Jordan* (con un programa en *BASIC*) se realiza en aproximadamente 1.5 segundos, y por el método de *Jacobi,* en 12 segundos. Sin embargo, el último da la solución exacta al sistema, mientras el de *Gauss-Jordan* no, debido, como ya vimos, a los errores de redondeo (la diferencia en este ejemplo es, no obstante, prácticamente nula).

El método consiste en despejar en la primera ecuación x_1, en la segunda x_2, en la tercera x_3 y así sucesivamente:

(2)

$$x_1 = \frac{19 - x_2 - x_3 - x_4}{10}$$

$$x_2 = \frac{11 - 2x_1 - x_3 + x_4}{5}$$

$$x_3 = \frac{13 + x_1 + x_2 - x_4}{4}$$

$$x_4 = \frac{17 - x_1 + x_2 + 2x_3}{5}$$

A continuación se toman valores aproximados para x_1, x_2, x_3... Si no se tiene una idea de cuáles pueden ser estos valores, basta



mas cuaciones les (II)

con hacer todos ellos iguales a la unidad (esto sólo hará que la convergencia sea más lenta). Nosotros tomaremos estos valores como valores iniciales, es decir:

$$x_1 = x_2 = x_3 = x_4 = 1$$

(Los valores exactos son $x_1 = 1$, $x_2 = 2$, $x_3 = 3$, $x_4 = 4$)

Acto seguido se sustituyen los valores iniciales en los segundos miembros de las ecuaciones (2):

$$x_1^1 = \frac{19 - 1 - 1 - 1}{10} = 1.6$$

$$x_2^1 = \frac{11 - 2 - 1 + 1}{5} = 1.8$$

$$x_3^1 = \frac{13 + 1 + 1 - 1}{4} = 3.5$$

$$x_4^1 = \frac{17 - 1 + 1 + 2}{6} = 3.16$$

Con lo que se obtienen los valores de x_1, x_2, x_3, x_4 correspondientes a la primera iteración (El superíndice 1 indica la iteración de la que se trata).

Estos valores se vuelven a sustituir en los segundos miembros de (2) para obtener la segunda iteración, y así sucesivamente, hasta llegar a una iteración 1 en que los valores resultantes sean iguales a los obtenidos en la iteración 1-1 en el número de cifras que deseemos.

Las iteraciones que siguen al problema ejemplo son:

Iteración 2:

$x_1^2 = 1.053$
$x_2^2 = 1.493$
$x_3^2 = 3.3083$
$x_4^2 = 4.03$

Iteración 3:

$x_1^3 = 1.0165$
$x_2^3 = 1.9236$
$x_3^3 = 2.8783$
$x_4^3 = 4.0094$

```
10 'SISTEMA DE ECUACIONES - JACOBI
20 'Versión 02.100286 - 3715 Bytes
30 'Juan Antonio Feberero Castejón
40''
50 CLEAR 200,60000!:SCREEN 0,,0:WIDTH 39:KEY OFF
60 DEFINT I-N:DEFSTR W
70 ON ERROR GOTO 10490
80 '
90 ' INTRO DATOS
100 '===========
110 '
120 CLS:LOCATE 4,10:PRINT "SISTEMAS DE ECUACIONES LINEALES",,,TAB(12);"METODO DE
 JACOBI
130 FOR I=1 TO 1000:IF INKEY$="" THEN NEXT
140 ON KEY GOSUB 10000,10010,10060,10160,10320
150 KEY(1)ON:KEY(5)ON
160 CLS:LOCATE 3,10:PRINT "<1> DATOS PROCEDENTES DE TECLADO",,,TAB(3);"<2> DATOS
 PROCEDENTES DE CINTA
170 W=INKEY$:IF W="" THEN 170
180 IF INSTR("12",W)=0 THEN 170
190 CLS:IF VAL(W)=2 THEN 370
200 '
210 'Datos de teclado
220 '----------------
230 '
240 INPUT "Nº DE ECUACIONES (Máx. 50)";N
250 DIM A(N,N+1),X(N),Y(N)
260 FOR I=1 TO N
270 FOR J=1 TO N
280 PRINT USING "A(##;##):";I;J;:INPUT A(I,J)
290 NEXT J
300 PRINT USING "B(##)    :";I;:INPUT A(I,N+1)
310 NEXT I
320 GOTO 560
330 '
340 'Datos de cinta
350 '--------------
360 '
370 PRINT "Preparar grabadora","Pulsar [RETURN]
380 WARCH="DATOS"
390 PRINT:PRINT "(TODOS LOS DATOS ALMACENADOS COMO CADE-NAS DE CARACTERES)",,,"N
OMBRE DEL ARCHIVO (RETURN)";:INPUT WARCH
400 OPEN WARCH FOR INPUT AS#1
410 PRINT:PRINT WARCH;": LO ENCONTRE"
420 LINE INPUT#1,WN
430 N=VAL(WN):DIM A(N,N+1)
440 FOR I=1 TO N
450 FOR J=1 TO N+1
460 IF EOF(1) THEN 510
470 LINE INPUT#1,W
480 A(I,J)=VAL(W)
490 NEXT J
500 NEXT I
510 CLOSE #1
520 '
530 'LIST. DATOS
540 '===========
550 '
560 CLS
570 KEY(2)ON:KEY(3)ON:KEY(4)ON
```

```
580 FOR I=1 TO N
590 FOR J=1 TO N+1
600 IF J=N+1 THEN PRINT USING "B(##)    =&";I;STR$(A(I,N+1)):GOTO 640
610 PRINT USING "A(##;##)=&";I;J;STR$(A(I,J))
620 IF CSRLIN=20 THEN 640
630 GOTO 700
640 W=INKEY$:IF W="" THEN 640
650 IF W=CHR$(30) AND I*J<N*N+N THEN 690
660 IF W=CHR$(31) AND J<N+2 THEN J=J-CSRLIN-20:IF J<-1 THEN I=I-2:GOTO 710 ELSE
690
670 IF W=CHR$(32) THEN 5040
680 GOTO 640
690 CLS
700 NEXT J
710 CLS:IF I<0 THEN I=0
720 NEXT I
5000 '
5010 'PROCESO
5020 '=======
5030 '
5040 CLS:LOCATE 15,11:PRINT "CALCULANDO...
5050 KEY(3)OFF:KEY(4)OFF
5060 FOR I=1 TO N
5070 A=0
5080 FOR J=1 TO N
5090 IF J=I THEN 5110
5100 A=A+A(I,J)
5110 NEXT J
5120 IF ABS(A)<ABS(A(I,I)) THEN 5170
5130 LOCATE 7,18:PRINT "POSIBLE NO CONVERGENCIA.",TAB(4)"SI DESEA SEGUIR PULSAR
[RETURN]
5140 W=INKEY$:IF W="" THEN 5140
5150 IF W=CHR$(13) THEN LOCATE 0,18:PRINT SPACE$(75):GOTO 5170
5160 END
5170 NEXT I
5180 FOR I=1 TO N:Y(I)=1:NEXT
5190 K=1
5200 FOR I=1 TO N
5210 X=A(I,N+1)
5220 FOR J=1 TO N
5230 IF J=I THEN 5250
5240 X=X-A(I,J)*Y(J)
5250 NEXT J
5260 X(I)=X/A(I,I)
5270 NEXT I
5280 K1=0
5290 FOR I=1 TO N
5300 IF X(I)=Y(I) THEN K1=K1+1
5310 Y(I)=X(I)
5320 NEXT I
5330 IF K1=N THEN 6040 ELSE K=K+1:GOTO 5200
6000 '
6010 'Resultados
6020 '----------
6030 '
6040 CLS
6050 PRINT "Nº DE ITERACIONES:";K:PRINT
6060 FOR I=1 TO N
6070 PRINT USING "x(##)=&";I;STR$(Y(I))
6080 IF CSRLIN=20 OR I=N THEN 6100
```

Quick

Los Joysticks más

QUICKSHOT IV
(3 en 1) Con mando
para deporte

QUICKSHOT IV (3 en 1)
Con mando de carreras

QUICKSHOT I MSX

QUICKSHOT I

QUICKSHOT VII - Portátil

QUICKSHOT IX
Preciso y sensible

Los QUICKSHOT comercializados por SVI-España, S. A. son los únicos que tienen la GARANTIA OFICIAL SVI.

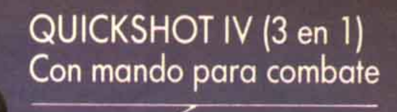
C&M

Shot®

vendidos del mundo.



QUICKSHOT II MSX
Con autodisparo



QUICKSHOT IV (3 en 1)
Con mando para combate



QUICKSHOT VII MSX
Portátil

QUICKSHOT II
Con autodisparo

Importador exclusivo SVI-España.



```
6090 GOTO 6160
6100 W=INKEY$:IF W="" THEN 6100
6110 IF W=CHR$(30) AND I<N THEN 6150
6120 IF W=CHR$(31) AND I>20 THEN I=I-CSRLIN-20:GOTO 6150
6130 IF W=CHR$(32) THEN 560
6140 GOTO 6100
6150 CLS
6160 NEXT I
10000 MOTOR:RETURN
10010 RETURN 560
10020 '
10030 'Modifica
10040 '--------
10050 '
10060 CLS
10070 INPUT "Dato a modificar: FILA";I
10080 INPUT "COLUMNA (0 para término independ.";J
10090 IF J=0 THEN J=N+1
10100 INPUT "NUEVO VALOR";A(I,J)
10110 RETURN 560
10120 '
10130 'GRABA CINTA
10140 '===========
10150 '
10160 CLS:PRINT "Preparar grabadora","Pulsar [RETURN]
10170 WARCH="DATOS"
10180 PRINT:PRINT "(TODOS LOS DATOS SE ALMACENAN COMO CA- DENAS DE CARACTERES)",
,,"NOMBRE DEL ARCHIVO (RETURN)";:INPUT WARCH
10190 OPEN WARCH FOR OUTPUT AS#1
10200 PRINT#1,STR$(N)
10210 FOR I=1 TO N
10220 FOR J=1 TO N+1
10230 PRINT#1,STR$(A(I,J))
10240 NEXT J
10250 NEXT I
10260 CLOSE #1
10270 RETURN 560
10280 '
10290 'Help
10300 '----
10310 '
10320 XP=POS(0):YP=CSRLIN
10330 FOR IHELP=1 TO 960:POKE 60000!+IHELP,VPEEK(IHELP):NEXT
10340 CLS
10350 PRINT "HELP",,"────"
10360 PRINT:PRINT "F1:Motor",,"F2:List. datos","F3:Modific. datos","F4:Grabar da
tos en cinta","F5:Help
10370 PRINT:PRINT:PRINT "LIS. DATOS",,STRING$(10,195)
10380 PRINT "Cursor:Página arriba/abajo","Barra espaciadora:Sale de List. Comien
-za proceso. Vuelve a List.",,,"En List. están activadas F1 a F5","En otro caso
están activadas F1, F2 Y  F5
10390 LOCATE 12,23:PRINT "Pulsa una tecla";
10400 IF INKEY$="" THEN 10400
10410 CLS
10420 FOR IHELP=1 TO 960:VPOKE IHELP,PEEK(60000!+IHELP):NEXT
10430 LOCATE XP,YP
10440 RETURN
10450 '
10460 'Errores
10470 '-------
10480 '
10490 CLS:LOCATE 7,12:PRINT "ERROR ";ERR;" EN LINEA ";ERL:STOP
```



LA REVISTA IMPRESCINDIBLE
PARA LOS USUARIOS DE LOS ORDENADORES PERSONALES MSX.

Una publicación mensual que ayuda a obtener el máximo partido a su ordenador.

MSX publica cada mes programas y juegos, además de reportajes sobre programación y la posibilidad de ganar premios realizando programas y otros temas siempre de gran interés.





ADEMAS, **beneficiese de un 15 % DE DESCUENTO sobre el precio real de suscripción**

PRECIO NORMAL DE SUSCRIPCION	USTED SOLO PAGA	AHORRO
~~3.600~~ PTAS.	3.060 PTAS.	15 %

APROVECHE AHORA esta irrepetible oportunidad para suscribirse a **MSX**. Envie **HOY MISMO** la tarjeta adjunta a la revista, que no necesita sobre ni franqueo. Deposítela en el buzón más cercano. Inmediatamente recibirá su primer ejemplar de **MSX** más el **REGALO**. Y así durante un año (12 números).



Bravo Murillo, 377
Tel. 733 79 69
28020 MADRID



En la sexta iteración, la diferencia entre la solución obtenida y el valor exacto es menor del 1 por ciento. En la iteración número 23, tenemos 10 cifras exactas, y en la número 34 obtenemos resultados iguales que en la 33 (coincidentes, además con el valor exacto).

En general, tendremos un sistema de **n** ecuaciones y **n** incógnitas. Decíamos al principio que podríamos estar seguros de la eficiencia del método si los elementos de la diagonal principal de la matriz de los coeficientes, predominaban sobre el resto de los elementos de la fila correspondiente. Más rigurosamente, podremos estar seguros de que la aplicación del método dará resultado si:

$$a_{ii} > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| \qquad (3)$$

donde a_{ii} es cada uno de los elementos de la matriz de los coeficientes. (i=fila, j=columna). En caso contrario, no podemos afirmar que el método sea convergente (aunque puede serlo).

Como puede observarse en el ejemplo, en la iteración k, la indeterminada x_i se calcula mediante la siguiente relación:

$$x_i^k = \frac{\sum\limits_{\substack{j=1 \\ j \neq i}} a_{ij} \, x_j^{k-1}}{a_{ii}} \qquad (4)$$

El diagrama 1 corresponde a una rutina de aplicación del método de *Jacobi.*

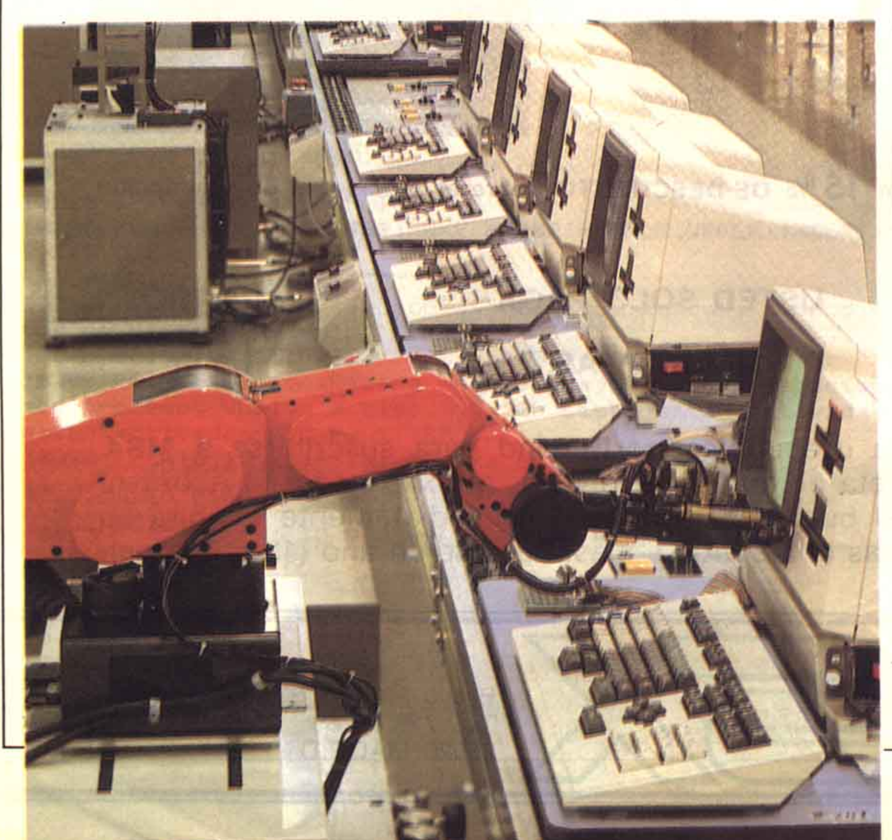
Método de Gauss-Seidel

Este método es, en todo, similar al de *Jacobi,* pero introduce la siguiente mejoría:

Partiendo del conjunto de valores iniciales elegido, se calcula la primera iteración para x_1 (x_1^1). Este resultado se toma como nuevo valor para x_1. Así, la serie de valores de $x_1, x_2, \ldots x_n$ para calcular la primera iteración de x_2 no está compuesta por $x_1^0, x_3^0, x_4^0, \ldots x_n^0$, sino por $x_1^1, x_3^0, x_4^0, \ldots x_n^0$. El nuevo valor x_1^2 obtenido para x_2 se toma para calcular x_3^1 ($x_1^1, x_2^1, x_4^0, \ldots x_n^0$), y así sucesivamente.

De esta forma se puede conseguir disminuir considerablemente el número de iteraciones necesarias para obtener una serie de valores $x_1^m, x_2^m, \ldots x_n^m$ que coincidan, en el número de cifras requerido, con los de la iteración precedente.

En el ejemplo anterior se obtienen los siguientes resultados en las primeras iteraciones:



```
10 'SIST. DE ECUACIONES - GAUSS-SEIDEL
20 'Versión 02.100286 - 3875 Bytes
120 CLS:LOCATE 4,10:PRINT "SISTEMAS DE ECUACIONES LINEALES",,,TAB(9);"METODO DE GAUSS-SEIDEL
5000 '
5010 'PROCESO
5020 '=======
5030 '
5040 CLS:LOCATE 15,11:PRINT "CALCULANDO...
5050 KEY(3)OFF:KEY(4)OFF
5060 FOR I=1 TO N
5070 A=0
5080 FOR J=1 TO N
5090 IF J=I THEN 5110
5100 A=A+A(I,J)
5110 NEXT J
5120 IF ABS(A)<ABS(A(I,I)) THEN 5170
5130 LOCATE 7,18:PRINT "POSIBLE NO CONVERGENCIA.",TAB(4)"SI DESEA SEGUIR PULSAR [RETURN]
5140 W=INKEY$:IF W="" THEN 5140
5150 IF W=CHR$(13) THEN LOCATE 0,18:PRINT SPACE$(75):GOTO 5170
5160 END
5170 NEXT I
5180 FOR I=1 TO N:Y(I)=1:NEXT
5190 K=1
5200 X(1)=A(1,N+1)
5210 FOR J=2 TO N
5220 X(1)=X(1)-A(1,J)*Y(J)
5230 NEXT J
5240 X(1)=X(1)/A(1,1)
5250 FOR I=2 TO N
5260 X(I)=A(I,N+1)
5270 FOR J=1 TO I-1
5280 X(I)=X(I)-A(I,J)*X(J)
5290 NEXT J
5300 IF I=N THEN 5340
5310 FOR J=I+1 TO N
5320 X(I)=X(I)-A(I,J)*Y(J)
5330 NEXT J
5340 X(I)=X(I)/A(I,I)
5350 NEXT I
5360 K1=0
5370 FOR I=1 TO N
5380 IF X(I)=Y(I) THEN K1=K1+1
5390 Y(I)=X(I)
5400 NEXT I
5410 IF K1=N THEN 6040 ELSE K=K+1:GOTO 5200
6000 '
```

Iteración 1:

$x_1^1 = 1.6$
$x_2^1 = 1.56$
$x_3^1 = 3.79$
$x_4^1 = 4.09$

Iteración 2:

$x_1^2 = 0.956$
$x_2^2 = 1.8776$
$x_3^2 = 2.9359$
$x_4^2 = 3.96556$

Iteración 3:

$x_1^3 = 1.022093$
$x_2^3 = 1.997096$
$x_3^3 = 3.0134056666665$
$x_4^3 = 4.0003023$

En la iteración número 4, la diferencia entre el resultado obtenido y el valor exacto es menor del 1 por ciento, y en la número 5, menor del 1 por mil.

En la iteración número 8 se obtienen 8 cifras exactas, y en la número 16, el resultado es exacto.

También aquí se puede asegurar la convergencia si se cumple la condición (3).

El valor x_i en la iteración k es, ahora:

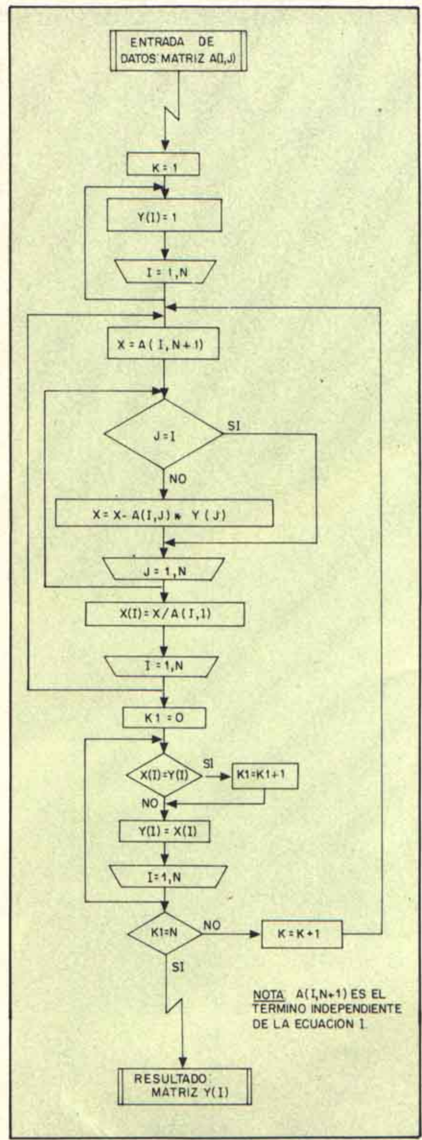


Diagrama 1. Método de Jacobi.

(5)

$$x_i^k = \frac{\sum_{j=1}^{i-1} a_{ij} x_j^k + \sum_{j=i+1}^{n} a_{ij} x_j^{k-1}}{a_{ii}}$$

La rutina del diagrama 2 es un desarrollo para resolver un sistema de ecuaciones lineales por este método.

A continuación se listan 2 programas correspondientes a cada uno de los dos métodos. Ambos son iguales al programa que se listaba en el número anterior de MSX Magazine, salvo en las líneas: 1Ø, 2Ø, 12Ø, y 5ØØØ a 6ØØØ.

En el segundo programa, que corresponde al método de *Gauss-Seidel*, sólo se han listado las líneas que son diferentes del primero. Por tanto basta con salvar el primer programa en cinta magnetofónica y, posteriormente, sustituir las líneas correspondientes al método *Gauss-Seidel*. (El hecho de no agrupar todos los métodos en un sólo programa es bueno, sobre todo si se va a manejar un número elevado de ecuaciones).

Recordemos cómo utilizar el programa.

Al correr el programa, se presentan dos opciones en pantalla:

1. Datos procedentes del teclado.
2. Datos procedentes de cinta.

Es interesante, fundamentalmente si se tiene un sistema con un elevado número de ecuaciones, guardar los datos en cinta magnetofónica y, posteriormente, utilizar la opción 2. Además, esto permitirá repetir el proceso con varios métodos para su posible comparación.

Se ha introducido una página de ayuda que se obtiene pulsando la tecla *F5*.

Pulsando *F1* se activa el motor de la grabadora. *F2* permite, en cualquier momento, listar los elementos de la matriz ampliada (con la ayuda del cursor). Estos se listarán automáticamente una vez se haya terminado de introducir los datos. A partir de este momento, y pulsando la barra espaciadora, comenzará la ejecución del programa. Si, una vez obtenidos los resultados, se vuelve a pulsar la barra espaciadora, vuelve a listarse la matriz ampliada, y así sucesivamente.

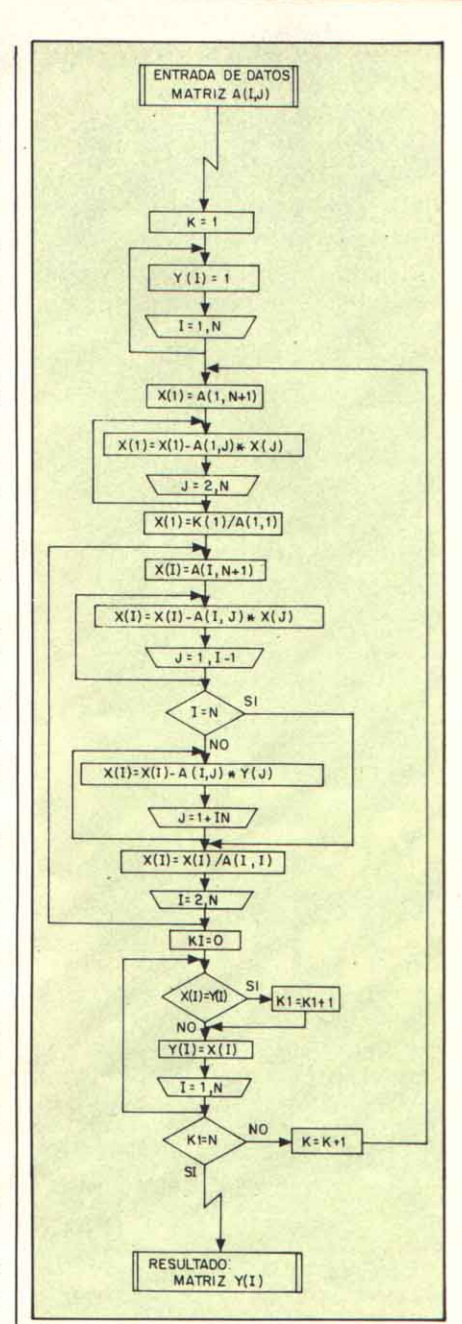
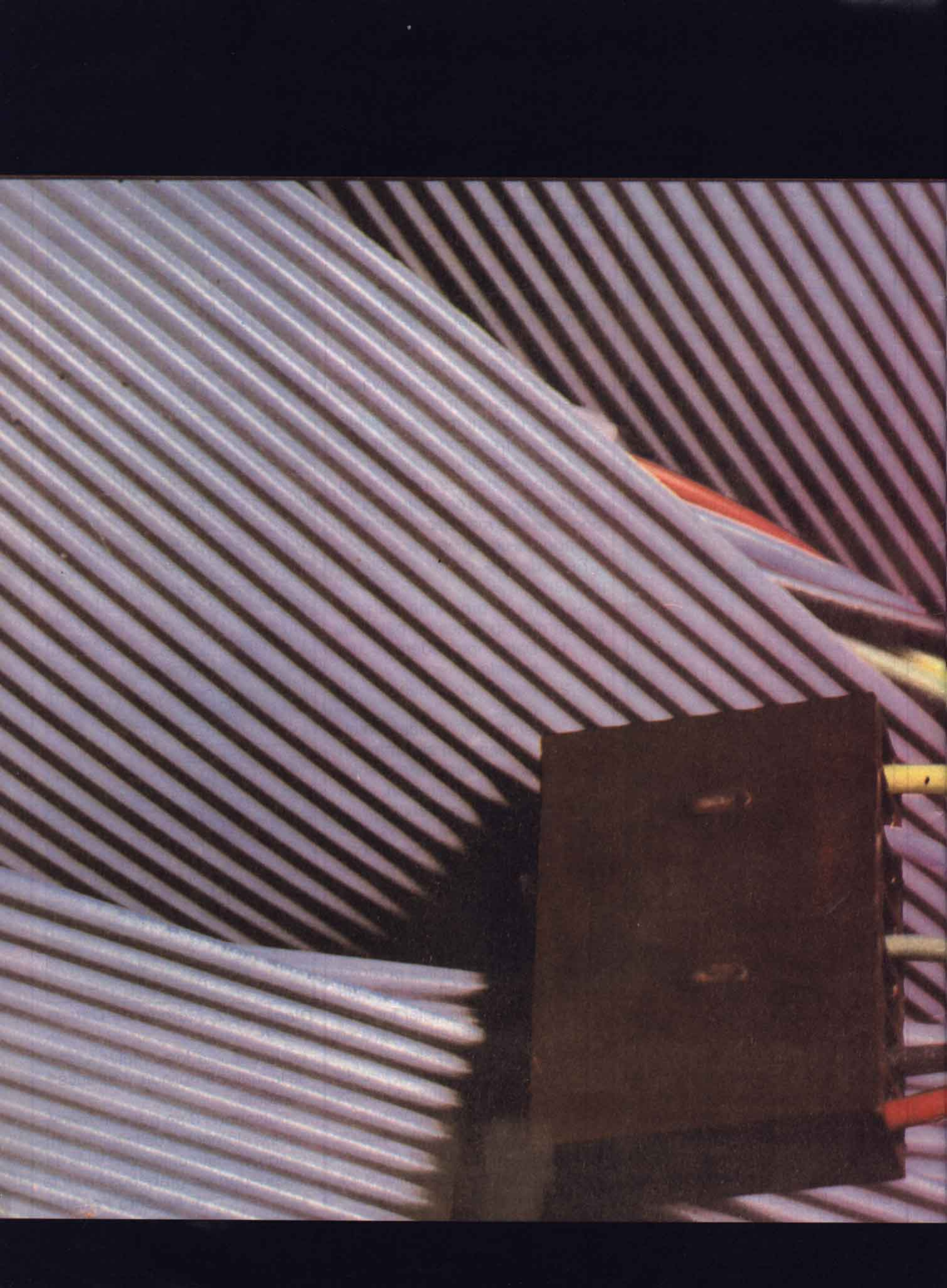


Diagrama 2. Método de Gauss-Seidel.

Pulsando *F3* se pueden modificar los elementos de la matriz ampliada, y, finalmente, con *F4* los datos pueden ser grabados en cinta.

J. Antonio Feberero





Llegados a este punto, vamos a cambiar un poco la orientación de esta serie de artículos. Ya hemos visto cómo se pueden utilizar las rutinas de la *ROM* en los ordenadores MSX y hemos abordado algunas instrucciones del Z80. Sin embargo, a la hora de desarrollar nuestras propias rutinas (programas) en código máquina, es necesario conocer el repertorio de instrucciones del microprocesador *Z80*, para poder elegir en cada caso la instrucción adecuada a nuestras necesidades. Pero no os asustéis: no se trata de memorizar la tabla de instrucciones que publicamos anteriormente, sino de comprender cómo funciona un grupo de instrucciones. En esta ocasión vamos a enfrentarnos con las instruccio-

Código Máquina

Esperamos que os pique el gusanillo del código máquina. Sin embargo, para poder programar en condiciones es necesario conocer bien el repertorio de instrucciones del Z80. Así pues, cambiaremos un poco la orientación de esta sección, analizando las instrucciones por grupos, y posteriormente abordaremos más seriamente la programación en código máquina con ejemplos concretos.

nes de carga, aritméticas y lógicas de 8 *bits*. Al decir «de 8 *bits*", nos referimos a que estas instrucciones funcionan sobre registros simples *(A, B, C, D, E, H, L)* y no sobre pares de registros *(BC, DE, HL)* ni registros *(IX, IY)*.

LD r,r': *r* y *r'* pueden ser *B,C,-D,E,H,L* o *A*. La operación se representa como *r <--- r'*, que quiere decir: el contenido del registro r' pasa al registro r. Veamos un ejemplo:

Supongamos que en el registro E tenemos *OFH* y en el *C, 030H*. Si ejecutamos la instrucción *LD C,E* haremos que *E* y *C* contengan lo mismo: *OFH*. Hemos de insistir en que el contenido del registro «origen» (del que sale el dato, en este caso el registro*E*) no cambia en una operación de carga entre registros, y el contenido del registro "destino" (el registro al cual llega el dato, en este caso el registro *C*) se hace igual al del registro origen.

LD r,n:r representa, igual que en el caso anterior, un registro simple, mientras que n representa un número de 8 *bits* (comprendido entre *OOH* y *OFFH*, ó 0 y 255 decimal). Se representa como *r <---n*, y quiere decir que el registro *r* se carga con el número *n*. Así, si tenemos en el registro *H* el número *12H* y ejecutamos *LD H,OFAH*, el registro *H* contendrá ahora *OFAH*.

LD r,(HL): se representa como *r <--- (HL)*. Evidentemente, al ser *HL* un par de registros (16 *bits*), no puede ser que el contenido de *HL* pase a un registro de 8 *bits*. Pero he de haceros notar que *HL* está entre paréntesis. Para explicar esto, introduciremos el concepto de «puntero».

Como sabéis, el *Z80* puede trabajar sobre 65536 direcciones de memoria, numeradas de 0 a 65535. Esto es así porque el bus que decodifica las direcciones tiene dieciséis líneas (dieciséis *bits*). Fijaros que los registros dobles y los pares de registros también tienen dieciséis *bits*. Pues bien, el *Z80* está preparado para poder usar un registro doble o un par de registros para decodificar una posición de memoria. Lo que se hace es considerar que el contenido del registro es una dirección de memoria, y ponerlo en el bus de direcciones. De este modo, decimos que la instrucción *LD B,(HL)* carga el registro B con el dato AL CUAL APUNTA *HL*. Veamos un ejemplo:

Supongamos que *HL* contiene *027DH*. Si consideramos esto como una dirección y miramos el dato que hay en esa dirección, podemos encontrarnos con, digamos, *6CH*. Si ejecutamos la instrucción de nuestro ejemplo *LD B,(HL)*, el *Z80* pone en el bus de direcciones *HL*, esto es *027DH*, y toma el dato que hay en esa dirección (es decir, *6CH*) y lo guarda en el registro *B*. El contenido de *HL* no varía con esta operación.

Para desarrollar nuestros propios programas en código máquina, es necesario conocer el repertorio de instrucciones del Z80.

LD r,(IX+d): *d* representa un desplazamiento a partir de la dirección a la que apunta el registro doble *IX*, y puede estar entre 127 y −128. Por ejemplo, *LD L,(IX+12)* hace que el procesador sume 12 al registro *IX*, y ponga el resultado de esta suma en el bus de direcciones (¡OJO!: el registro *IX* se queda como estaba; la suma no se guarda) y coja el dato que hay en la dirección a la que «apunta» *IX+12*, para pasarlo al registro *L*. Se representa como r *<--- (IX+d)*. Por supuesto, el contenido de la dirección de memoria de la cual se toma el dato tampoco varía.

LD r,(IY+d): exactamente igual que la anterior, pero con el registro *IX* en lugar del *IY*.

LD (HL),r: se representa como *(HL) <---r*. Se trata del caso inverso a *LD r,(HL)*. La diferencia está en que el dato se toma de un registro y se almacena en la memoria, en la dirección a la que «apunta» *HL*.

LD (IX+d),r: caso inverso de *LD r,(IX+d)*. Se representa como *(IX+ d) <---r*.

LD (IY+d),r: igual que el anterior, pero con el registro *IY*. Se representa por *(IY+d) <---r*.

LD (HL),n: se representa como *(HL) <---n*. Es igual que *LD (HL),r* con la diferencia de que el «origen» del dato no es un registro, sino el número que acompaña a la instrucción *(n)*. Nuevamente *HL* no cambia en esta posición, ni *IX* ni *IY* en las anteriores.

LD A,(BC): se representa como *A <---(BC)*. Como veis, es parecida a *LD r,(HL)*. Se diferencia en que se utiliza el par de registros *BC* como puntero, y en que el registro «destino» SOLO PUEDE SER EL A. Es decir, no existe *LD L,(BC)*.

Nuevamente, *BC* no resulta alterado.

LD A,(nn):nn representa un número de dieciséis *bits*. Este número es el que se pone en el bus de direcciones para coger el dato al cual «apunta» y pasarlo al registro A.

LD (BC),A, LD (DE), A y *LD (nn),A* son las operaciones opuestas a las tres anteriores.

SERVICIO DE EJEMPLARES ATRASADOS

ESTOS SON LOS EJEMPLARES DE MSX MAGAZINE APARECIDOS EN EL MERCADO CON UN RESUMEN DE SU CONTENIDO



Núm. 1
¿Qué es el MSX? Su BASIC, periféricos, programas, software.



Núm. 2
Generación de sonido. MSX-DOS, el ordenador por dentro, programas, noticias.



Núm. 3
Los joysticks, 256 caracteres programables, Z80 corazón de león, compro/vendo/cambio.



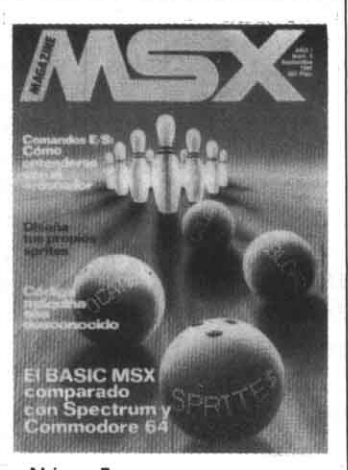
Núm. 4
Las comunicaciones entre ordenadores, la jerga informática, trucos, rincón del lector.



Núm. 6
Los 8 magníficos (test gigante), el bus de expansión, los misterios de la grabación, programas.



Núm. 8
Compact Disc, el periférico del futuro. Test: Dynadata DPC-200. Continuamos con la memoria de video. Libros, software, programas, trucos.



Núm. 5
Comandos de entrada/salida, el BASIC MSX comparado con Spectrum y Commodore 64, Código Máquina.



Núm. 7
Analizamos el Generador de Sonido. Aplicaciones matemáticas con el ordenador. La memoria de video. Trucos, noticias.



Núm. 9
Características técnicas del Compact Disc. Tratamiento de datos. Test: Quick Disk. Trucos, libros, noticias, programas.

PARA HACER SU PEDIDO, RELLENE ESTE CUPON, HOY MISMO Y ENVIELO A MSX MAGAZINE BRAVO MURILLO, 377. Tel. 733 96 62 - 28020 MADRID

Ruego me envien los siguientes números atrasados ....

al precio de **250** ptas. cada uno. Cuyo importe abonaré:

☐ POR CHEQUE ☐ CONTRA REEMBOLSO ☐ CON MI TARJETA DE CREDITO

☐ AMERICAN EXPRESS ☐ VISA ☐ INTERBANK

Número de mi tarjeta ....

Fecha de caducidad ....

NOMBRE ....

DIRECCION ....

POBLACION .... C.P. ....

PROVINCIA ....

Existen cuatro instrucciones más en las que no entraremos: *LD A,I, LD I,A, LD A,R y LD R,A.* Estas operaciones se refieren al refresco de memoria y al manejo de interrupciones, y son utilizadas por el diseñador del aparato, por lo que normalmente no las necesitaremos.

En todas las operaciones comentadas hasta ahora, los flags no resultan afectados de modo alguno, continuando como estaban antes de la operación.

Hasta aquí las instrucciones de carga de 8 *bits.* Vamos ahora con las aritméticas y lógicas.

ADD r:r puede ser de nuevo cualquiera de los registros *B,C,-D,E,H,L o A.* Se representa como A <---*A + r,* y quiere decir que se suma el contenido de *A* con el contenido del registro *r,* y el resultado SE GUARDA EN A. El registro r no resulta afectado. Esta operación afecta a todos los flags.

ADD n: análoga a la anterior, si bien la suma es del acumulador con el número que sigue a la instrucción. Se representa como *A <--- A + n,* y afecta igualmente a todos los flags.

ADD (HL): suma el acumulador con el número de 8 *bits* contenido en la posición de memoria a la que «apunta» el par de registros *HL.* Se representa por *A <--- A + (HL)* y afecta a todos los flags.

ADD (IX+d): d representa un desplazamiento (ver instrucciones de carga). Se suma el acumulador con el número contenido en la posición de memoria a la que apunta *IX+d.* El registro *IX* no resulta alterado, ni tampoco el contenido de memoria. Se representa como *A <--- (IX+d),* y afecta a todos los flags.

ADD (IY+d): igual que la anterior, pero con el registro doble *IY.* Se representa como *A <--- (IY+d).*

ADC s: puede ser cualquier registro simple *(B,C,D,E,H,L,A),* un número *n* o el dato apuntado por *(HL), (IX+d)* o *(IY+d).* Es igual que la suma normal *ADD* pero al resultado se le suma el *bit* de acarreo. Es decir, si el acarrero era cero, el resultado será el mismo que con *ADD,* y si era uno, será una unidad mayor. Se representa como *A <--- A + s + CY,* conde *CY* representa a la bandera de acarreo (flag de acarreo); afecta a todos los flags.

SUB s: s representa lo mismo que en *ADC.* La operación es restar el acumulador menos *s,* y guardar el resultado en el acumulador *(A <--- A − s).* Afecta a todos los flags.

El Z80, puede trabajar sobre 65536 direcciones de memoria, numeradas de 0 a 65535.

SBC s: igual que *SUB,* si bien al resultado se le resta el *bit* de acarreo. Se representa como *A <--- A − s − CY,* y afecta a todos los flags.

AND s, OR s, XOR s: estas tres instrucciones efectúan la correspondiente operación lógica *(AND, OR* o *XOR)* entre el acumulador y *s* (*s* representa lo mismo que en las instrucciones anteriores) y guarda el resultado en el acumulador *A <--- A AND s, A <--- A OR s, A <--- A XOR s).* Afectan a todos los flags, y hay que destacar que el flag de acarreo queda siempre a cero, ya que por la propia naturaleza de la operación no puede darse nunca un acarrero.

CP s: resta el acumulador menos *s,* pero NO GUARDA EL RESULTADO. Tanto el acumulador como *s* permanecen inalterados, pero no así los flags. Esta operación se utiliza para comparar dos números y actuar según sean iguales o según cuál sea mayor. Se representa como *A - s.*

INC r: r puede ser *B,C,D,E,H,L,* o *A.* Afecta a todos los flags excepto el acarreo, y se representa como *r <--- r + 1.*

DEC r; igual que la anterior, pero *r <--- r − 1* (resta 1 en vez de sumar 1).

INC (HL), INC (IX+d), INC (IY+d): suman uno a la posición de memoria a la que «apunta» el correspondiente puntero. Los registros *HL, IX* e *IY* no resultan alterados. Se representan como *(HL) <--- (HL)+ 1; (IX+d) <--- (IX+d)+1; (IY+d) <--- (IY+d)+1.* Alteran a todos los flags excepto el de acarreo.

DEC (HL), DEC (IX+d), DEC (IY+ d): igual a los anteriores pero restando uno en lugar de sumando uno. Se representan por *(HL) <--- (HL)−1; (IX+d) <--- (IX+d)−1; (IY+ d) <---(IY+d)−1.*

Para comprender todo esto, veamos algunos ejemplos:

Supongamos que tenemos dos grupos de cinco números, uno de ellos a partir de la dirección *0C000H* y otro a partir de la dirección *0C100H,* y queremos sumar el primero de un grupo con el primero del otro, el segundo de uno con el segundo del otro, y así hasta el quinto, y guardar los resultados en memoria a partir de *0C200H;* podemos hacer lo siguiente:

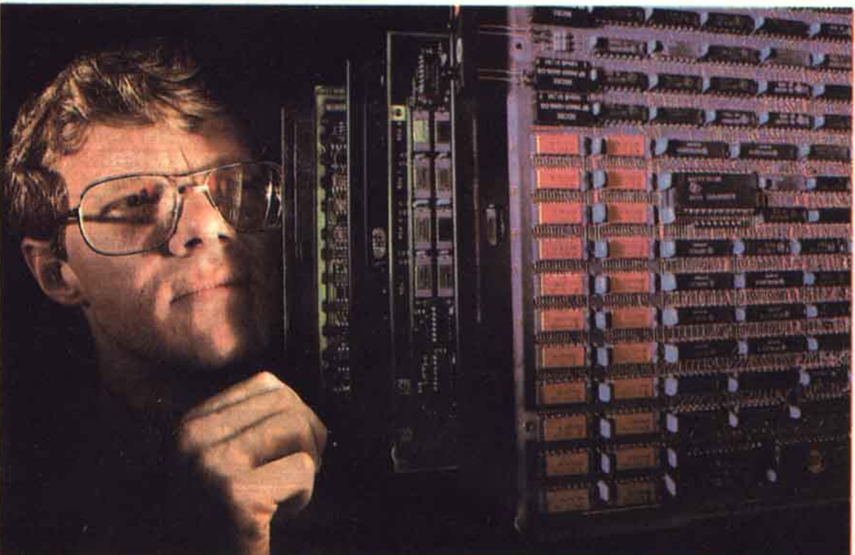
— *LD IX,0C000H:* Carga puntero con dirección primer grupo.

— *LD IY,0C100H:* Carga puntero con dirección segundo grupo.

— *LD HL,0C200H:* Carga puntero con dirección resultado.

— *LD A,(IX+0):* Carga primer sumando grupo 1.





— *ADD A,(IY+0):* Suma primer sumando grupo 2.

— *LD (HL),A:* Almacena primer resultado.

— *INC (HL):* Actualiza puntero de resultados.

— *LD A,(IX+1):* Carga siguiente sumando grupo 1.

— *ADD A,(IY+1):* Carga siguiente sumando grupo 2.

— *LD (HL),A:* Almacena siguiente resultado.

— *INC HL:* Actualiza puntero resultados.

Y así hasta cinco. Como véis, este proceso para sumar dos tablas de 100 números ocuparía mucha memoria, pero sirve como ejemplo del uso de los punteros IX, IY y HL. Veámos ahora otra forma más corta y útil de hacer lo mismo:

— *LD BC,0C000H:* Carga puntero con dirección primer grupo.

— *LD HL,0C100H:* Carga puntero con dirección segundo grupo.

— *LD DE,0C200H:* Carga puntero con dirección resultados.

— *LD A,05H:* Carga número de números a sumar.

— *LAZO: LD (0BFFFH),A:* Y lo guarda en memoria.

— *LD A,(BC):* Carga sumando grupo 1.

— *ADD A,(HL):* Suma sumando grupo 2.

— *LD (DE),A:* Guarda resultado en memoria.

— *LD A,(OBFFFH):* Carga el contador de sumas.

— *DEC A:* Lo actualiza (contador=contador—1).

— *CP 00H:* Mira si el contador es cero.

— *CP 00H:* Mira si el contador es cero.

— *JP NZ,LAZO:* Continúa hasta que el contador sea cero.

— *RET:* Vuelve al programa principal.

El efecto de la instrucción *JP NZ* es verificar el flag de cero *(Z)* y, si este es cero (en este caso, debido a que el acumulador no sea cero al llegar a *CP 00H)* salta a la dirección marcada con la etiqueta *LAZO,* ignora el alto y pasa a la siguiente instrucción.

Como véis, este método resulta bastante más corto, y además es muy flexible, pues cambiando el dato de la cuarta instrucción *(LD A,05H)* podemos sumar bloques de 1 a 255 números.



Deseo intercambiar programas con usuarios de MSX. Llamar a Fernando al Tel.: (91) 469 43 44.

Vendo consola de juegos Atari junto a 9 cartuchos de juegos a muy buen precio. También cambio videojuegos en cinta para ordenadores de la marca SV-318 y 328. Llamar a Jorge al Tel.: (93) 331 93 65.

Vendo SVI-328, data cassette SV-904, TV en B/N, 4 manuales y 25 programas, todo por 60.000 ptas. Ponerse en contacto con Ramón Llorens Moreno. C/ Guipúzcoa, 8. 08018 Barcelona, o llamar al Tel.: (93) 308 62 53.

Me gustaría contactar con usuarios de MSX de Elche (Alicante) para intercambio de programas, material, etc. Dirigirse a Vicente Rubio Mañas.

Vendo ordenador SVI-328 de 32K ROM y 80K RAM, cassette SV-904, con sus respectivos manuales, joystick y 5 programas, por 40.000 ptas. Dirigirse a Igor. Avda. Madariaga, 15, 6-C. 48014 Bilbao, o llamar al Tel.: (94) 447 87 34 (tardes).

Vendo Spectravideo 728 con garantía, adquirido en Octubre de 1985. Adjunto manual y programas. Precio total, 40.000 ptas. Escribir a Ramón Domingo Mustaros, Calabria, 168. 08015 Barcelona. Teléfono: 421 95 25.

Vendo programas 100% en código máquina a un precio módico. Escribir a José C. Martínez Acosta. Zamora, 99. Vigo o llamar al Tel.: (986) 41 45 25.


GANE
7.000 PTAS.
todos los meses
PARTICIPANDO
EN NUESTRO
CONCURSO
MSX Magazine premiará cada mes los programas que nos hagan llegar nuestros lectores.
Para participar en este concurso abierto, todo aficionado a los ordenadores con este estándar deberá hacer llegar a la redacción de la revista el listado, un cassette y un texto explicativo.
Entre todos los programas que recibamos cada mes, serán seleccionados para su publicación aquellos que reúnan los siguientes criterios:
— Originalidad de la aplicación.
— Simplicidad del método de programación.
La única condición para participar en el concurso será que los programas no hayan sido publicados previamente en ninguna revista.
Enviar vuestros programas a: MSX Magazine
C/Bravo Murillo, 377 - 5.° A 28020 MADRID
BANCO
DE
ESPAÑA
MIL
pesetas
2E1554046
CINCO MIL
pesetas
14457150

Rincón _del lector

ESCRIBIR POR ESCRIBIR

Estimados amigos:

Escribo la presente con motivo de un artículo que apareció en su revista hermana «Ordenador Popular» (Número 33-febrero 1986) titulado «MSX: estándar para la educación». En este artículo aparecen frases como

«fracaso del software».

«La compatibilidad no fue tan absoluta».

«...poco potentes...», etc...

No sé si estas críticas son fundadas, ya que no me considero un experto en la materia, pero me voy a permitir contradecir ésta y otras muchas críticas que están apareciendo en las revistas especializadas.

Respecto al software, después de dos años en el mercado, no podemos esperar que tenga la misma cantidad de programas que marcas como Spectrum o Commodore.

La potencia, si la comparamos con IBM PC o Apple Macintosh, es de risa, pero no podemos salirnos del tipo de ordenador. Todos los que tenemos un MSX sabemos que es un ordenador doméstico, y hay que compararlo con este tipo de aparatos, y entonces ya podemos hablar de ventajas y desventajas según con cuál lo comparamos. Los MSX, desde luego, tienen sus ventajas y desventajas que son, en mi opinión, fundamentalmente tres:

a) Aparecieron cuando el mercado ya estaba saturado.

b) No presentan ninguna novedad importante. Me refiero con esto a los chips que presentan, no a la forma de explotarlos que es muy eficiente a pesar de la tercera desventaja.

c) Alto precio para la configuración base y sus periféricos.

Este último punto es el más influyente en la no consolidación de los MSX en nuestro país.

Si eliminamos el factor precio y suministramos a nuestro ordenador los periféricos necesarios, podemos contar con un aparato bastante potente. Así podemos trabajar con CP:M, uno de los sistemas operativos más prestigiosos para ordenadores de 8 bits, y no es de extrañar que Commodore lanzara su último modelo con este 8.0. Lo que nunca se ha pretendido es que un ordenador MSX llevara la contabilidad de una gran empresa, ni que poseyera una base de datos con todos los clientes de unos grandes almacenes.

Hay que ser serios, y si los MSX no han triunfado «plenamente», es por que la competencia ha defendido su mercado, y no critico esto último, ya que en estos factores está basada nuestra sociedad.

En resumen, y mirándolo fríamente, los MSX no son tan malos como «ellos» lo quieren pintar, por lo que aconsejo a mis compañeros que no se preocupen de las críticas y que ellos mismos hagan la comparación. Os aseguro que os tranquilizaréis.

J. M. Rial

El artículo en cuestión, desde mi punto de vista, está escrito sin fundamento, de forma grosera y peyorativa. Huelga decir que el autor de semejante refrito, no ha tenido (ni tendrá) un MSX, ni sabe las posibilidades de este sistema, ni hasta dónde puede llegar, porque lo que está claro es que seis millones de japoneses no se pueden equivocar. Pero ellos no son los únicos que han adoptado la nueva norma. Una empresa europea y una americana, Philips y Spectravídeo respectivamente, también se han apuntado a la estandarización de los ordenadores personales, lo cual es un paso muy importante a todos los niveles.

Indudablemente, MSX no tuvo el auge que tuvo Sinclair o el Commodore 64, si se me apura mucho. Es un ordenador que, desgraciadamente, ha llegado tarde al mercado, pero sin ningún tipo de triunfalismos, como destaca ese artículo. Es más, en la actualidad hay unos 20 fabricantes de MSX, entre los que se han repartido el mercado del estándar, lo que significa que el número de ordenadores de una casa será ínfimo comparado con lo vendido por Sinclair, pero si sumamos el número total de ordenadores MSX que hay en circulación podemos llegar a conclusiones más concretas.

Pero no me voy a extender en lo referente a ese tema y me centraré en tu carta, puesto que como bien dices al final de ella, lo mejor que puede hacer cualquier usuario es comparar ordenadores.

Para empezar, no se puede decir que el software fracasó. Que apareciera lentamente es una cosa, pero de ahí al fracaso media un abismo. Esto, además, resulta lógico, ya que desarrollar programas para un aparato nuevo lleva su tiempo. No se puede, ni se debe improvisar de la noche a la mañana como ciertos personajillos pretenden.

En lo que respecta a la compatibilidad, podemos decir una cosa bien clara en la que los fabricantes —estoy seguro— me apoyarán; el 95% de los MSX son compatibles. ¿Qué ocurre con el otro 5%? Simplemente que hay algunos que por sus características peculiares destacan del resto, resultando algo superiores. Lo que sí vamos a dejar claro es que TODO el software que hemos probado, así como un gran número de programas de aplicación y sistemas operativos (a excepción del CP/M que incorpora el Spectravídeo X'press) corre debidamente en el resto de los ordenadores.

Sin embargo, lo que sí me causó verdadera estupefacción es que se diga que son poco potentes. Tal es el asombro con el que leí el artículo que sólo me queda decir una cosa: ¿qué opinión le mereció al autor de dicho artículo la aparición en España del ZX-81 o el mismo VIC-20, precursores del Spectrum 48K y del Commodore 64K?

Juan Arencibia

COPIAR PANTALLAS EN LA IMPRESORA

¿Podrían decirme cómo puede pasarse una pantalla gráfica a la impresora?

¿Se pueden crear más de un intervalo con la instrucción ON INTERVAL GOSUB?

Salvador Mainé López
Cádiz

Para copiar pantallas a una impresora hace falta recurrir a una rutina en código máquina que publicaremos en meses sucesivos dentro de la sección de TRUCOS.

En cuanto a la instrucción ON INTERVAL GOSUB, ésta permite definir un sólo intervalo, por lo que es imposible realizar lo que indicas.



LIBROS CON ERRORES Y CASSETTE QUE NO CARGA

Tengo un problema en un programa de un libro que comentaron en el número 8 de MSX. Su título es, «El libro de los juegos para MSX» cuyo autor es Andrew Lacey.

Pues bien, he copiado el programa «Localizador» de la página 12, y al ejecutarlo me da el error "OVERFLOW IN 1Ø", lo he comprobado al máximo y no he cometido ningún error al copiarlo (lo he repasado varias veces y está como en el libro).

Por otro lado, me compré un cassette especial para ordenadores y no graba bien. Sin embargo, las cintas de juegos que poseo sí cargan correctamente. ¿Esto ocurre con los cassettes?

J. Torralbo
Barcelona

Es el primer caso que tenemos sobre un error en este libro y nos extraña, puesto que los controles a los que son sometidos los libros de Anaya son muy severos en este aspecto, por lo que sólo nos queda la opción de indicarle que se diriga a esa editorial y les exponga la duda.

Los cassettes, especialmente para ordenadores, suele ser en la mayoría de los casos un timo. Es preferible gastarse algo más y no correr el riesgo de que nos tomen el pelo. Por esta razón te sugerimos que utilices un aparato de la misma marca que tu ordenador, ya que los fabricantes de MSX también hacen sus propios periféricos, que aunque sean algo más caros suelen dar un resultado inmejorable.

INFORMACION SOBRE EL CD-ROM

Debido a mi interés por el CD-ROM, derivado de vuestro artículo «CD-ROM, base de datos en casa» desearía obtener más información, tanto técnica como económica sobre estos aparatos, sobre todo los vídeo-disc.

Acacio Dais Fernandes
Barcelona

La información del Compact Disc la recopilamos gracias a Philips y a Pioneer, dos grandes empresas que nos facilitaron todos los datos acerca de las características de dichos aparatos. En el momento de escribir aquel artículo, tuvimos la suerte de ser invitados de excepción cuando Philips realizó una demostración conjunta con su ordenador de la segunda generación, que era un prototipo y un Compact Disc (en la feria de Sonimag del año 85).

Todavía es pronto para saber el auténtico alcance y las verdaderas posibilidades de los ordenadores MSX con un periférico de esa clase, pero no cabe la menor duda que se está trabajando en este tema y que se está avanzando a pasos agigantados.

DIRECTOR:
Juan Arencibia.
COORDINADOR EDITORIAL:
J. Ignacio Rey.
COLABORADORES:
Octavio López, Angel Zarazaga, Teresa Aranda, Ricardo García.
DISEÑO:
Benito Gil y Ricardo Segura.
Editada por:
PUBLINFORMATICA, S.A.
C/ Bravo Murillo, 377 - 5.º A
Tel.: 733 74 13
28020 Madrid.
Telex 48877 OPZXE
PRESIDENTE:
Fernando Bolín.
DIRECTOR EDITORIAL REVISTAS DE USUARIOS:
Juan Arencibia.
DIRECTOR DE VENTAS:
Antonio González.
JEFE DE PRODUCCIÓN:
Miguel Onieva.
SERVICIO AL CLIENTE:
Julia González.
Tel.: 733 79 69
DIRECCION, REDACCION Y ADMINISTRACION:
C/ Bravo Murillo, 377 - 5.º A.
Tel.: 733 74 13
28020 Madrid.
PUBLICIDAD EN MADRID:
Emilio García.
PUBLICIDAD EN BARCELONA:
Lidia Cendros.
C/ Pelayo, 12.
Tel.: (93) 301 47 00 Ext. 27-28.
08001 Barcelona.
Depósito Legal: M. 16.755-1985
Imprime: G. Velasco, S. A.
Distribuye:
S.G.E.L. Avda. Valdelaparra, s/n.
Alcobendas (Madrid).
DISTRIBUIDORES:
VENEZUELA: SIPAM, S.A. Avda. República Dominicana, 541
ARGENTINA: DISTRIBUIDORA INTERCONTINENTAL BUENOS AIRES.

El P.V.P. para Ceuta, Melilla y Canarias, incluido servicio aéreo será de 300 ptas. sin I.V.A.

SUSCRIPCIONES:
Rogamos dirija toda la correspondencia relacionada con suscripciones a:
MSX
EDISA: Tel. 415 97 12
C/López de Hoyos, 141-5.º
28002 MADRID
(Para todos los pagos reseñar solamente MSX)
Para la compra de ejemplares atrasados dirijanse a la propia editorial
MSX
C/Bravo Murillo, 377-5.º A
Tel. 733 74 13 28020 MADRID

Si deseas colaborar en MSX remite tus artículos o programas a Bravo Murillo 377, 5.º A. 28020 Madrid. Los programas deberán estar grabados en cassette y los artículos mecanografiados.
A efectos de remuneración, se analiza cada colaboración aisladamente, estudiando su complejidad y calidad.

micro byte PRESENTA... MSX

SENSACIONALES PROGRAMAS EN CARTUCHO Y CASSETTE

FLIGHT PATH 737.



Colócate a los mandos de un jet comercial. Disponemos de control total sobre los mandos del avión, y puedes escoger entre 6 niveles de dificultad.

**P.V.P.: CART. 3.490 pts.
CASS. 1.900 pts. 32K.**

FRUITY FRANK



Tu jardín ha sido invádido por monstruos de fruta madura. La única forma de combatirlos es lanzarles fruta fresca del jardín.

P.V.P.: CASS. 1.900 pts. 32K.

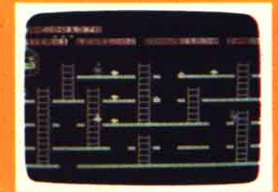
SPARTAN X



Son muchos los peligros que te acechan. Ten los reflejos bien despiertos, pon tus fuerzas en estado de alerta, y a luchar.

P.V.P.: CASS. 1.900 pts. 32K.

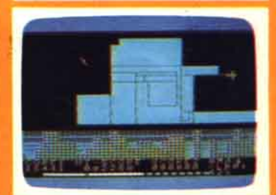
CHUCKIE EGG



Debes recoger los huevos antes de que nazcan los pollitos y se cóman el maiz. Pero ojo con el Pato Loco.

P.V.P.: CASS. 1.900 pts. 32K.

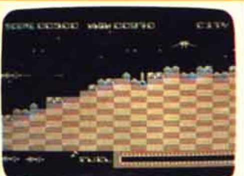
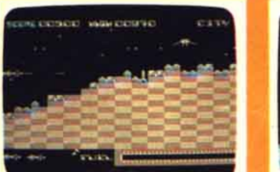
NIGHT FLIGHT



Con tu pequeño avión, debes ir dando luz a la noche, hasta que el cielo esté de nuevo azul. Date prisa en realizar tu misión, de lo contrario...

**P.V.P.: CART. 2.900 pts.
CASS. 1.900 pts. 16K**

STAR AVENGER

Imagina el juego de batalla más rápido que jamás hayas visto. Piensa además, en los más excitantes gráficos y sus 5 niveles de dificultad. Todo ello es Star Avenger.

P.V.P.: CASS. 1.900 pts. 32K.

GYRO ADVENTURE



Ponte a los mandos de tu helicóptero y combate a los enemigos que se enfrentan a ti. Podrás mover el helicóptero en todas direcciones, mantenerlo en el aire y disparar. **P.V.P.: CART. 2.900 pts. CASS. 1.900 pts. 16K.**

SUPER CROSS FORCE



Sólo queda una esperanza para la supervivencia ante el ataque de los malvados Morpul. Tú podrás atacarles, con tus naves dispuestas en paralelo o en diagonal.

P.V.P.: CART. 2.900 pts. 16K.

JUMP LAND



Tu mayor obsesión han sido siempre los pasteles, y por ello, te has visto envuelto en situaciones complicadas que has salvado gracias a tus reflejos.

P.V.P. CASS. 1.900 pts. 16K.

ROGER RUBBISH



Los perversos contaminadores de planetas están llenando nuestra galaxia de residuos nucleares. Roger Rubbish es el más famoso recogedor de basuras espaciales.

P.V.P.: CART. 2.900 pts. 16K.

FRUIT PANIC



Un día, Walky, para divertirse se fué al país de los gatos. ¿Cuánta fruta podrá comerse Walky?

P.V.P.: CASS. 2.000 pts. 16K

DIZZY BALLOON



En este mundo hay seres voladores y atacan cuerpo a cuerpo. Si los haces explotar, se irá abriendo el cielo y tendrás la oportunidad de escapar.

P.V.P.: CASS. 2.000 pts. 32K.

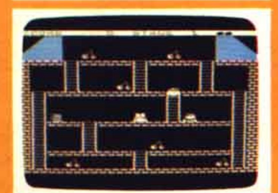
CASTLE COMBAT



El castillo galáctico, ha caido bajo la dominación de los Tyrones. Tu nave STAR DUSTER, está prepareda para el combate. ¿Te atreves?

P.V.P.: CART. 2.900 pts. 16K.

NICK NEAKER



Cuando estás dormido, muchas cosas suceden a tu alrededor. Algunos objetos de tu casa toman vida, como en el caso de la zapatilla NICK.

**P.V.P.: CART. 2.900 pts.
CASS. 1.900 pts. 16K.**

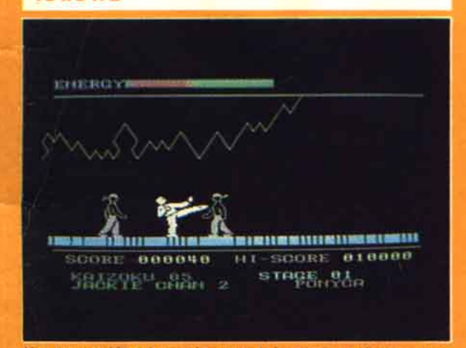
CHAMP



Champ es un completo Ensamblador/Monitor para tu MSX. Champ te permite escribir y trazar programas en código máquina con el mínimo esfuerzo.

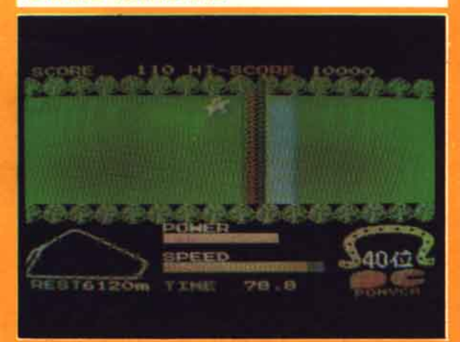
**P.V.P.: CART. 3.890 pts.
CASS. 2.400 pts. 32K.**

KARATE



Has conseguido entrar en la cueva de los piratas y ahora comienzan tus problemas. Los murciélagos gigantes, moradores de estas cuevas pueden chuparte la sangre. Cuando te encuentres con los piratas, deberás enfrentarte a ellos con tu depurado estilo de Kárate. **P.V.P.: CART. 3.490 pts. CASS. 1.900 pts. 32K.**

GRAND NATIONAL



Si te gustan las carreras de caballos, no te quedes como un espectador, participa. Ahora puedes correr con tu caballo, en la más prestigiosa carrera del mundo, el GRAND NATIONAL.

P.V.P.: CASS. 2.000 pts. 32K.

ENVIENOS A **MICROBYTE**

P.º Castellana, 179, 1.º - 28046 Madrid

Nombre
Apellidos
Dirección
Población
D.P. Teléfono

ENVIOS GRATIS

JUEGO	Cart.	Cass.	Precio	TOTAL

PRECIO TOTAL PESETAS

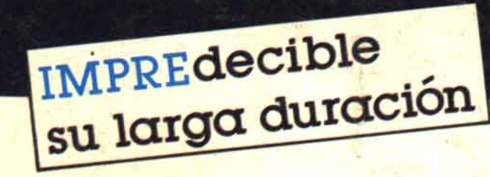
Incluyo talón nominativo ☐
Contra-Reembolso ☐

Pedidos por teléfono **91 - 442 54 33 / 44**













GP-50 •	La pequeña 40 cps. Papel normal con interface paralelo, serial y Spectrum	17.990 ptas.
GP-700 *	La de color 50 cps. 7 colores. 80 columnas. Tracción y fricción. Papel de 10 pulgadas	64.990 ptas.
SP-1.000 *	La programable 100 cps.24 cps en alta calidad 96 cart. programables en RAM. Introductor hoja a hoja.♦	64.990 ptas.
SP-1.000AS	La programable 100 cps.24 cps en alta calidad con interface RS-232. Introductor hoja a hoja.♦	59.900 ptas.
MP-1.300AI	La polivalente 300 cps, 60 cps en alta calidad, interface paralelo y RS-232. Introductor hoja a hoja.♦&	119.900 ptas.
BP-5.200 *	La de oficina 200 cps, 106 en alta calidad.Buffer 4K.Carro de 15".Tracción y fricción.♦	199.900 ptas.
BP-5.420 *	La más rápida 420 cps. 106 cps en alta calidad. Buffer de 18K. Paralelo y RS-232.♦	339.900 ptas.

Interfaces: Serie RS-232C, Spectrum, IBM, COMMODORE, MSX, QL, Apple Macintosh, HP-IB
♦ Introductor automático de documentos opcional.
& Kit de color opcional.

* con interface paralelo
• con interface Spectrum

Nota: I.V.A. 12%, no incluido en los precios arriba indicados

Avda. Blasco Ibáñez, 116
Tel. (96) 372. 88. 89
Telex 62220 - 46022 VALENCIA

Muntaner, 60-2.º-4.ª
Tel. (93) 323. 32. 19
08011 BARCELONA

Agustín de Foxá, 25-3.º-A
Tels. (91) 733. 57. 00 - 733. 56. 50
28036 MADRID

